

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Казанский национальный исследовательский технологический университет»**

**(ФГБОУ ВО «КНИТУ»)**

Кафедра ИСУИР

Направление подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Тема курсовой работы (проекта) по дисциплине «Визуальное программирование» «Разработка прототипа информационной системы «Учет телекомпанией стоимости прошедшей в эфире рекламы»

**КУРСОВОЙ ПРОЕКТ**

Заведующий кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Кирпичников А. П.)

Руководитель проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Мангушева А. Р.)

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Шукуров Э. Р.)

Нормоконтролер\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Мангушева А. Р.)

Казань 2018 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc515395026)

[ЗАДАНИЕ 4](#_Toc515395027)

[ЛИСТ НОРМОКОНТРОЛЯ 5](#_Toc515395028)

[ГЛАВА 1. РАЗРАБОТКА ПРОЕКТА ИНФОРМАЦИОННОЙ СИСТЕМЫ 6](#_Toc515395029)

[1.1. Описание предметной области 6](#_Toc515395030)

[1.2. Описание глоссария проекта и действующих лиц 6](#_Toc515395031)

[1.3. Определение вариантов использования 7](#_Toc515395032)

[1.4. Создание диаграммы классов 18](#_Toc515395033)

[1.5. Описание ER-модели базы данных 20](#_Toc515395034)

[ГЛАВА 2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ 22](#_Toc515395035)

[2.1. Подключение к базе данных 23](#_Toc515395036)

[2.2. Вход в систему 24](#_Toc515395037)

[2.3. Ведение информации о клиенте 25](#_Toc515395038)

[2.4. Ведение информации о рекламе 31](#_Toc515395039)

[2.5. Просмотр личной статистики 34](#_Toc515395040)

[2.6. Ведение информации о сотрудниках 36](#_Toc515395041)

[2.7. Изменение стоимости за рекламу 39](#_Toc515395042)

[2.8. Просмотр статистики 41](#_Toc515395043)

[2.9. Выход из системы 43](#_Toc515395044)

[ЗАКЛЮЧЕНИЕ 45](#_Toc515395045)

[СПИСОК ЛИТЕРАТУРЫ 46](#_Toc515395046)

# ВВЕДЕНИЕ

Цель данного курсового проекта ­­­­- разработка прототипа информационной системы «Учет телекомпанией стоимости прошедшей в эфире рекламы» с помощью СУБД SQL Server и языка программирования C# в среде Visual Studio.

Информационная система (ИС) – это система, предназначенная для манипулирования информацией.

Тема данного курсового проекта актуальна в современном мире, т.к. СМИ играют значимую роль и в развитии рыночной экономики, и в повседневной жизни.

Для выполнения поставленной цели необходимо выполнить следующие задачи:

1. описание предметной области;
2. описание глоссария проекта и действующих лиц ИС;
3. определение вариантов использования разрабатываемой системы;
4. создание диаграммы классов;
5. описание ER-модели базы данных;
6. разработка прототипа информационной системы.

## ЗАДАНИЕ

на курсовую работу (проект) студенту кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема работы (проекта)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Исходные данные к проекту:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание расчетно-пояснительной записки (включая перечень подлежащих разработке вопросов, включая вопросы стандартизации и контроля качества) ВВЕДЕНИЕ. ГЛАВА1. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ. 1.1.Описание предметной области 1.2. Описание глоссария проекта и действующих лиц. 1.3. Определение вариантов использования 1.4. Создание диаграммы классов 1.5. Описание ER-модели базы данных ГЛАВА 2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ 2.1. Подключение к базе данных 2.2. Вход в систему 2.3. Ведение информации о клиенте 2.4. Ведение информации о рекламе 2.5. Просмотр личной статистики 2.6. Ведение информации о сотрудниках 2.7. Изменение стоимости за рекламу 2.8. Просмотр статистики 2.9. Выход из системы. ЗАКЛЮЧЕНИЕ. СПИСОК ЛИТЕРАТУРЫ.

Перечень графического материала (схемной документации) **\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Консультанты по проекту (с указанием относящихся к ним разделов)\_\_\_\_\_\_

Дата выдачи задания "\_\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г.

Руководитель проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

## ЛИСТ НОРМОКОНТРОЛЯ

1. Лист является обязательным приложением к пояснительной записке дипломного (курсового) проекта.

2. Нормоконтролер имеет право возвращать документацию без рассмотрения в случаях:

- нарушения установленной комплектности,

- отсутствие обязательных подписей,

- нечеткого выполнения текстового и графического материала.

3. Устранение ошибок, указанных нормоконтролером, обязательно.

ПЕРЕЧЕНЬ

замечаний и предложений нормоконтролера по курсовому проекту студента

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Группа, инициалы, фамилия)

|  |  |  |
| --- | --- | --- |
| Лист  (страница) | Условное обозначение  (код ошибок) | Содержание замечаний и предложений со ссылкой на нормативный документ, стандарт или типовую документацию |
|  |  |  |

Дата «\_\_\_» \_\_\_\_\_\_\_\_\_20\_\_\_г.

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (инициалы, фамилия)

# ГЛАВА 1. РАЗРАБОТКА ПРОЕКТА ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 1.1. Описание предметной области

Телекомпания «ВИД» имеет в своей структуре рекламный отдел, главной задачей которого является размещение рекламы во время показа передач.

Клиент обращается в рекламный отдел. Рекламный агент согласует с ним все необходимые условия, а именно: время показа рекламы, и продолжительность. После чего клиенты предоставляют информацию о себе: фамилию, имя, отчество, паспортные данные, номер банковского счета и номер телефона. Агент сохраняет все вышеперечисленные данные в систему.

Рекламный агент получает определенный процент за рекламу.

Стоимость одной минуты за рекламу является переменной величиной и её устанавливает глава отдела. Также глава отдела может добавлять новых сотрудников, и соответственно, удалять ныне работающих.

Действующие лица:

* рекламный агент – занимается оформлением договоров с клиентами;
* глава отдела – занимается оформлением договоров с рекламными агентами, изменяет стоимость услуги.

## 1.2. Описание глоссария проекта и действующих лиц

В глоссарий проекта были включены следующие термины:

1. Клиент – лицо, заказывающее рекламу;
2. Эфир – время для показа передач.

Выделим действующие лица информационной системы:

* рекламный агент – занимается оформлением договоров с клиентами;
* глава отдела – занимается оформлением договоров с рекламными агентами, изменяет стоимость услуги.

## 1.3. Определение вариантов использования

На основе потребностей действующих лиц выделим следующие варианты использования:

* + - * Рекламный агент:

- ведение информации о клиенте;

- просмотр личной статистики;

- ведение информации о рекламе.

* + - * Глава рекламной службы:

- ведение информации о сотрудниках;

- изменение стоимости за рекламу;

- просмотр статистики.

* + - * Варианты использования для всех лиц:

- вход в систему;

- выход из системы.

На рисунке 1 приведена схема вариантов использования, на которой в виде стрелок показаны ассоциации между действующем лицом, главой отдела, и вариантами использования.

На рисунке 2 приведена схема вариантов использования для рекламного агента.

На рисунке 3 приведена схема вариантов использования для всех действующих лиц.

На рисунках 4-6 описаны структуры вариантов использования «Ведение информации о клиенте», «Ведение информации о сотрудниках» и «Ведение информации о рекламе» соответственно. Каждый из этих вариантов использования включают по три потока, которые может выбрать пользователь. Например, при выполнении «Ведение информации о клиенте» рекламный агент может выбрать одно из действий: добавить нового клиента в систему, редактировать данные о клиенте или удалить клиента из системы.

Далее приведем описание вариантов использования для действующих лиц.



Рисунок 1 – Диаграмма вариантов использования для главы отдела.

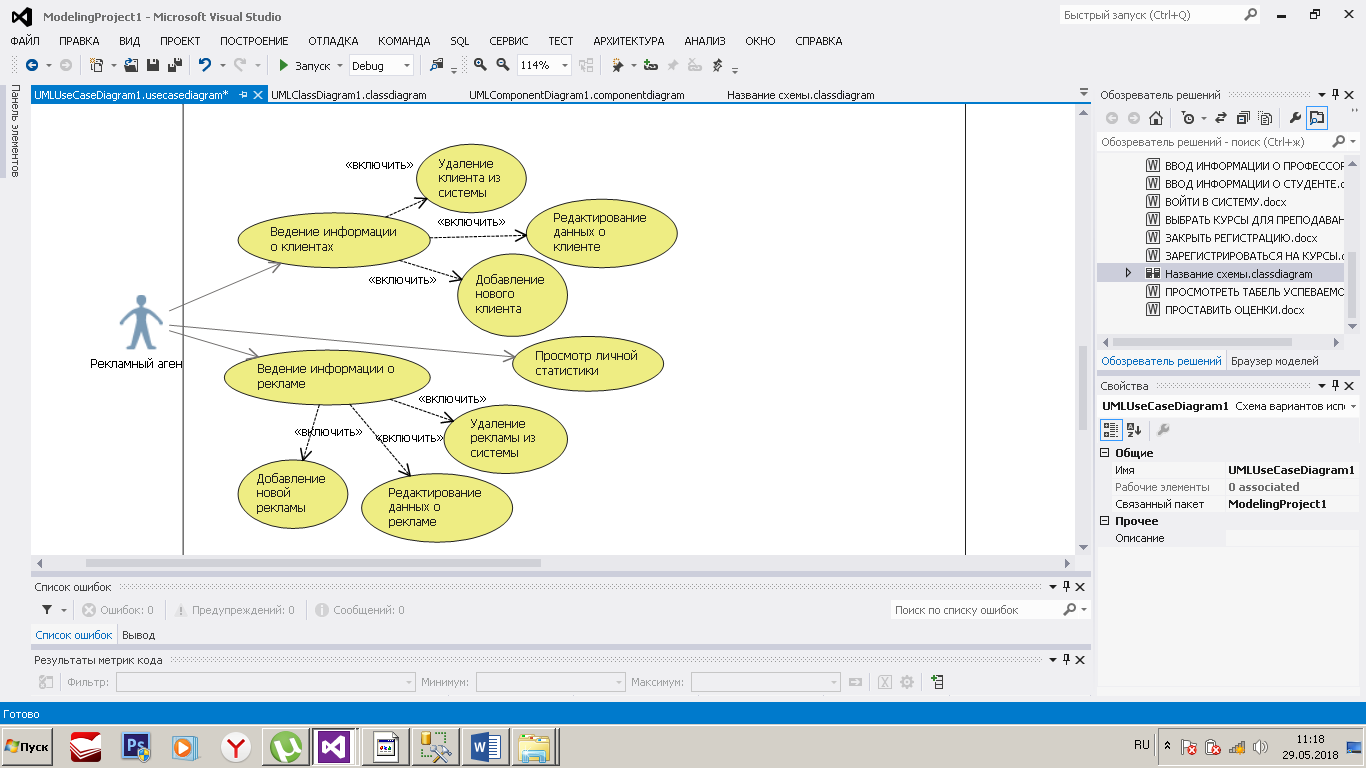


Рисунок 2 – Диаграмма вариантов использования для рекламного агента.

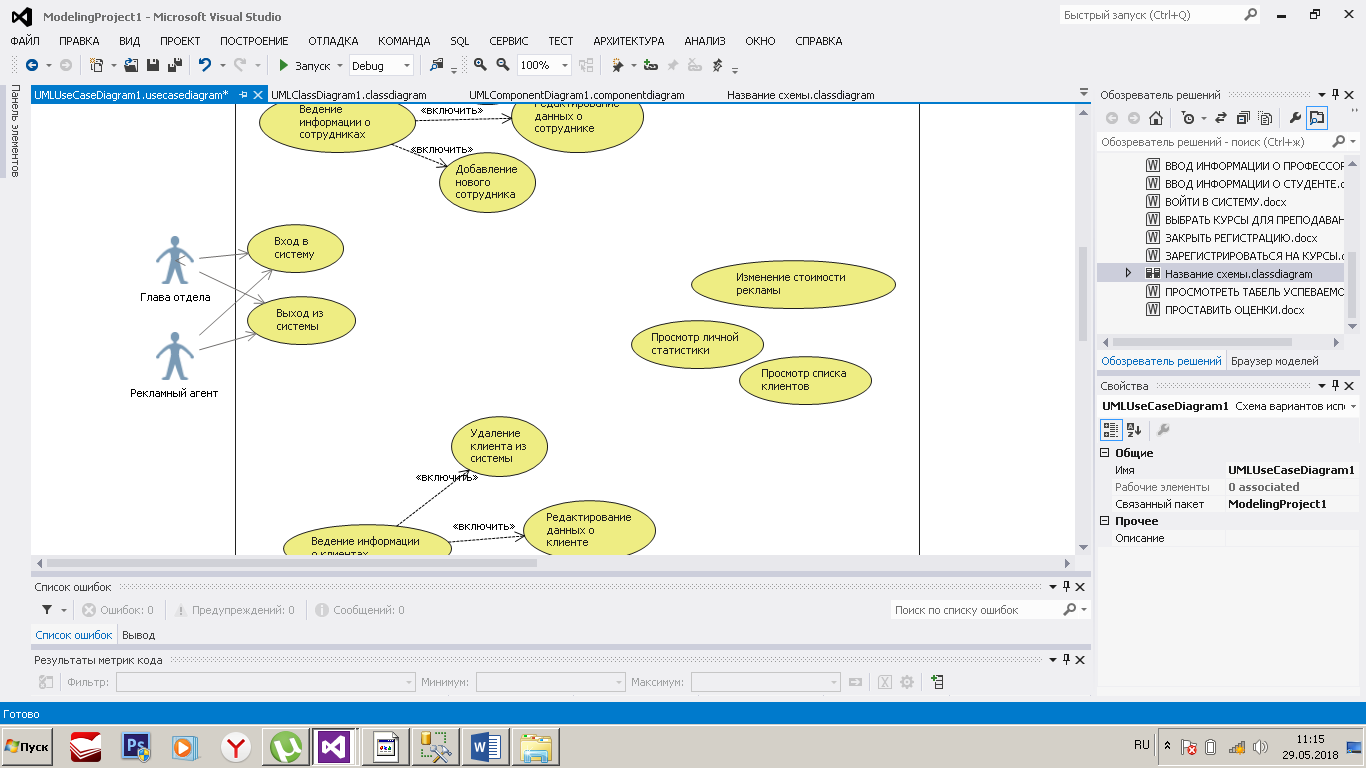


Рисунок 3 – Диаграмма вариантов использования для всех действующих лиц.

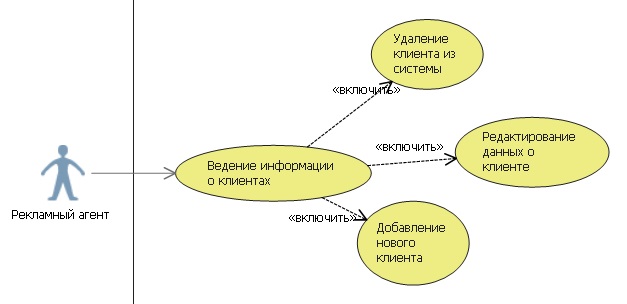


Рисунок 4 – Вариант использования «Ведение информации о клиентах».

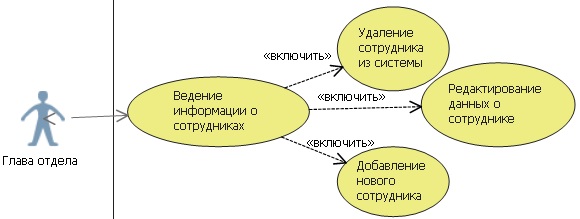


Рисунок 5 – Вариант использования «Ведение информации о сотрудниках».



Рисунок 6 – Вариант использования «Ведение информации о рекламе».

Вариант использования: «Вход в систему»

*Краткое описание:*

Данный вариант использования описывает вход пользователей в систему.

*Основной поток событий:*

Данный вариант использования начинает выполняться, когда кому-либо из действующих лиц потребуется пройти авторизацию в системе.

1. система открывает окно для ввода логина и пароля;
2. пользователь заполняет поля и нажимает кнопку «Войти».
3. система проверяет введенные данные и предоставляет пользователю доступ в систему.

*Альтернативный поток событий:*

*Неверный логин или пароль:*

3. При вводе неверного логина или пароля система выводит соответствующее сообщение, после чего пользователь может снова попытаться ввести логин и пароль.

Для дальнейших вариантов использования должно выполняться предусловие: действующее лицо (пользователь) должно войти в систему.

Вариант использования: «Выход из системы»

*Краткое описание:*

Данный вариант использования описывает выход пользователя из системы.

*Основной поток событий:*

Вариант использования начинает выполняться, когда пользователю необходимо выйти из системы после завершения работы в ней или для предоставления входа в нее другому пользователю.

1. пользователь нажимает правой кнопкой мыши по изображению в главном меню или может закрыть его стандартным способом (см. пункт 4);
2. если пользователь использовал изображение в главном меню, то система предоставляет ему кнопку «Выйти»;
3. после чего пользователь нажимает на соответствующую кнопку;
4. система выводит сообщение;
5. если пользователь нажимает на кнопку «Да», то система закрывает текущую форму и открывается окно для ввода логина и пароля.
6. пользователь может закрыть приложение, нажав на кнопку «Выйти»
7. система выводит сообщение;
8. если пользователь нажимает на кнопку «Да», то приложение полностью закрывается.

Вариант использования: «Ведение информации о клиенте»

*Краткое описание:*

Данный вариант использования описывает процесс создания, редактирования и удаления пользователем записей о клиентах.

*Основной поток событий:*

Вариант использования начинается после того, как пользователь выбрал категорию «Внести изменения» из меню «Клиенты». Система открывает список клиентов, с которыми он заключил договор. Пользователь может выбрать одного клиента из списка двойным нажатием по таблице. В таком случае форма для заполнений записей автоматически заполнится соответствующими данными. Тогда рекламный агент может выполнить удаление, редактирование или добавление записи. Также пользователь может создать новую запись о клиенте, вручную прописав данные в форме.

*«Добавление нового клиента»*

1. система предоставляет пользователю окно с формами для ввода данных о клиенте: ФИО, паспортные данные, номер счета и телефон;
2. рекламный агент вводит все необходимые данные и нажимает на кнопку «Добавить»;
3. система запрашивает подтверждение пользователя;
4. система сохраняет введенные данные в каталоге клиентов и выводит соответствующее сообщение.

*«Редактирование данных о клиенте»*

1. рекламный агент выбирает клиента двойным нажатием по записи в таблице;
2. форма автоматически заполняется информацией о выбранном клиенте, пользователь редактирует все необходимые ему данные и нажимает на кнопку «Изменить»;
3. система запрашивает подтверждение пользователя;
4. пользователь подтверждает;
5. система сохраняет введенные изменения в каталоге клиентов и выводит соответствующее сообщение.

*«Удаление клиента из системы»*

1. рекламный агент выбирает клиента двойным нажатием по записи в таблице;
2. пользователь нажимает на кнопку «Удалить»;
3. система запрашивает подтверждение пользователя;
4. пользователь подтверждает удаление;
5. система удаляет данные выводит соответствующее сообщение.

*Альтернативные потоки:*

*«Добавление нового клиента»*

1. Некоторые данные нового клиента совпадают с данными уже существующего клиента в базе. Система выводит соответствующее сообщение. Если совпадают паспортные данные, банковские реквизиты или номер телефона, то в этом случае выясняется, что клиент ранее обслуживался рекламной службой и ввод данных будет отменен. Если совпадают только ФИО, то сообщение может быть проигнорировано и ввод данных о новом клиенте будет подтвержден.

2. Не все необходимые поля для ввода были заполнены. В этом случае система выведет соответствующее сообщение. Пользователь может заполнить необходимые поля или прервать вариант использования.

*«Редактирование данных о клиенте»*

1. Новые данные клиента совпадают с данными уже существующего клиента в базе. Если совпадают банковские реквизиты или номер телефона, то редактирование данных будет отменено. Если совпадают только ФИО, то сообщение может быть проигнорировано и ввод данных о новом клиенте будет подтвержден.

2. Не все необходимые поля для ввода были заполнены. В этом случае система выведет соответствующее сообщение. Пользователь может заполнить необходимые поля или прервать вариант использования.

3. Пользователь не подтвердил изменение данных. Он может вернуться к редактированию записи или прервать вариант использования.

*«Удаление клиента из системы»*

После запроса о подтверждении пользователь не подтверждает удаление. В этом случае удаления данных не происходит.

Вариант использования: «Изменение стоимости минуты рекламы»

*Краткое описание:*

Данный вариант использования описывает редактирование стоимости минуты рекламы.

*Основной поток событий:*

1. пользователю открывается окно с формой для ввода стоимости и выбора передачи;
2. пользователь задает нужное значение для выбранной передачи;
3. система запрашивает подтверждение пользователя;
4. пользователь подтверждает;
5. система сохраняет значение.

*Альтернативный поток:*

1. Пользователь не подтверждает редактирование. В этом случае цена остается прежней.
2. Пользователь ввел неверные данные, например, строковое значение. Выводится соответствующее сообщение.

Вариант использования: «Ведение информации о сотрудниках»

*Краткое описание:*

Данный вариант использования описывает процесс создания, редактирования и удаления главой рекламного отдела записей о сотрудниках.

*Основной поток событий:*

Вариант использования начинается после того, как пользователь нажал кнопку «Сотрудники». Система открывает список всех сотрудников. Выбрав сотрудника двойным кликом по записи в таблице, глава отдела может выполнить удаление или редактирование записи о нем. Также пользователь может создать новую запись о сотруднике.

*«Добавление сотрудника»*

1. система предоставляет пользователю окно с формами для ввода данных о сотруднике: ФИО, дата найма и процент, при этом логин и пароль создаются автоматически путем генерации случайных символов;
2. глава отдела вводит все необходимые данные и нажимает на кнопку «Добавить»;
3. система запрашивает подтверждение пользователя;
4. система сохраняет введенные данные в каталоге сотрудников.

*«Редактирование данных о сотруднике»*

1. глава отдела выбирает сотрудника двойным нажатием по записи в таблице;
2. форма автоматически заполняется соответствующей информацией о выбранном сотруднике и пользователь редактирует все необходимые данные и нажимает «Изменить»;
3. система запрашивает подтверждение пользователя;
4. пользователь подтверждает;
5. система сохраняет введенные изменения в каталоге сотрудников.

*«Удаление клиента из системы»*

1. глава отдела выбирает сотрудника двойным нажатием по записи в таблице;
2. пользователь нажимает на кнопку «Удалить»;
3. система запрашивает подтверждение пользователя;
4. пользователь подтверждает;
5. система удаляет данные.

*Альтернативные потоки:*

*«Добавление нового сотрудника»*

Не все необходимые поля для ввода были заполнены. В этом случае система выведет соответствующее сообщение. Пользователь может заполнить необходимые поля или прервать вариант использования.

*«Редактирование данных о сотруднике»*

1. Не все необходимые поля для ввода были заполнены. В этом случае система выведет соответствующее сообщение.

2. Пользователь не подтвердил изменение данных. Он может вернуться к редактированию записи или прервать вариант использования.

*«Удаление сотрудника из системы»*

После запроса о подтверждении пользователь не подтверждает удаление. В этом случае удаления данных не происходит.

Вариант использования: «Ведение информации о рекламе»

*Краткое описание:*

Данный вариант использования описывает процесс создания и удаления сотрудником личных записей о рекламе.

*Основной поток событий:*

Вариант использования начинается после того, как сотрудник выбрал «Просмотр рекламы» из выпадающего меню «Реклама». Система открывает список с добавленными рекламами.

Выбрав запись о рекламе двойным кликом по ней в таблице, сотрудник может выполнить удаление записи. Также пользователь может создать новую запись о рекламе.

*«Добавление данных о рекламе»*

1. система предоставляет пользователю окно с формами для ввода или выбора данных: ФИО и паспорт существующих в базе клиентов, название рекламы, передача, во время которой будет показана реклама, количество секунд и время показа;
2. сотрудник вводит все необходимые данные и нажимает на кнопку «Добавить»;
3. система запрашивает подтверждение пользователя;
4. система сохраняет введенные данные.

*«Удаление рекламы из системы»*

1. сотрудник выбирает запись о рекламе двойным нажатием по ней в таблице;
2. пользователь нажимает на кнопку «Удалить»;
3. система запрашивает подтверждение пользователя;
4. пользователь подтверждает;
5. система удаляет данные.

*Альтернативные потоки:*

*«Добавление новой рекламы»*

1. Некоторые данные совпадают с данными уже существующей рекламы в базе. Система выводит соответствующее сообщение. Если совпадают название рекламы и время её показа, то ввод данных будет отменен. Если совпадают другие данные, то сообщение будет проигнорировано и ввод данных о новой рекламе будет подтвержден.

2. Не все необходимые поля для ввода были заполнены. В этом случае система выведет соответствующее сообщение.

*«Редактирование данных о рекламе»*

1. Данные клиента совпадают с данными уже существующей рекламы в базе. Система выводит соответствующее сообщение. Если совпадают название рекламы и время её показа, то ввод данных будет отменен. Если совпадают другие данные, то сообщение будет проигнорировано и ввод данных о новой рекламе будет подтвержден.

2. Не все необходимые поля для ввода были заполнены. В этом случае система выведет соответствующее сообщение.

3. Пользователь не подтвердил изменение данных. Он может вернуться к редактированию записи или прервать вариант использования.

*«Удаление рекламы из системы»*

После запроса о подтверждении пользователь не подтверждает удаление. В этом случае удаления данных не происходит.

Вариант использования: «Просмотр личной статистики»

*Краткое описание:*

Данный вариант использования отображает личную информацию о сделках.

*Основной поток событий:*

Вариант использования начинается после того, как пользователь выбрал меню «Статистика». Система открывает список с добавленными рекламами, а также показывает количество сделок и доход сотрудника.

*Предусловие:*

Только сотрудник может имеет доступ к данному варианту использования.

Вариант использования: «Просмотр статистики»

*Краткое описание:*

Данный вариант использования отображает информацию о сделках.

*Основной поток событий:*

1. вариант использования начинается после того, как пользователь выбрал меню «Статистика»;
2. система открывает список с добавленными рекламами и данными о клиентах;
3. пользователь может выбрать из выпадающего списка отдельного сотрудника;
4. пользователь нажимает на кнопку «Поиск»;
5. система предоставляет все добавленные рекламы сотрудника, а также его доход.

*Предусловие:*

Только глава отдела может имеет доступ к данному варианту использования.

## 1.4. Создание диаграммы классов

Диаграмма классов - это схема, в которой имеются классы системы и связи между ними. Также на данной диаграмме показываются атрибуты классов их операции и ограничения, накладываемые на связи [1]. В рамках данного курсового проекта классы разделены на две категории:

- граничные классы (boundary classes) – это классы, которые находятся на границе системы и окружающей средой, примерами данных классов могут служить экранные формы или интерфейсы других систем [2];

- классы-сущности (entity classes) ­– классы, которые хранят в себе некую информацию, каждому такому классу соответствует таблица в базе данных (БД) [3].

На рисунке 7 показана диаграмма классов, которая состоит из трех граничных классов и шести классов - сущностей. Все граничные классы представляют экранные формы.

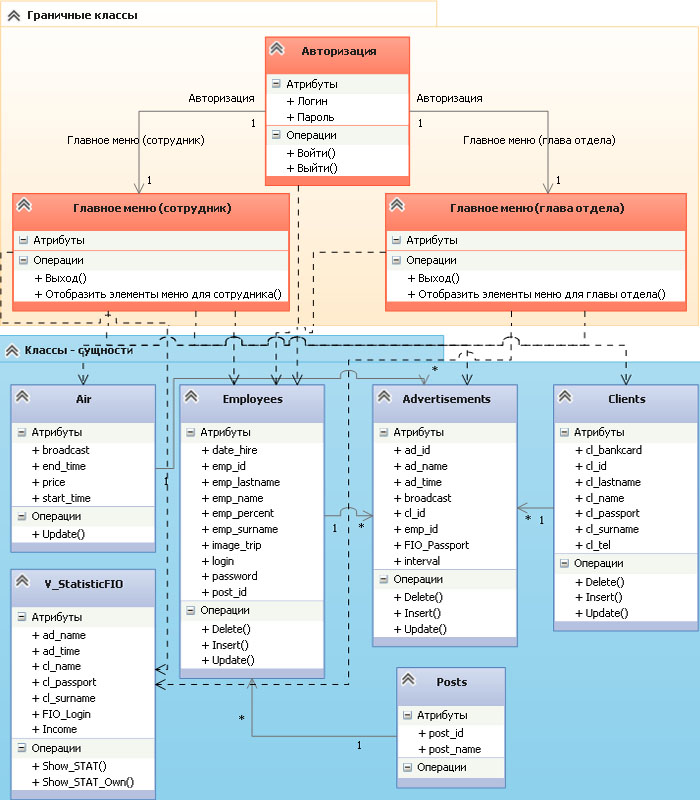


Рисунок 7 – Диаграмма классов.

Приведем описание одного из классов – сущностей.

Класс «Employees» содержит в себе следующие атрибуты: emp\_id (уникальный номер рекламного агента), emp\_surname (фамилия), emp\_name (имя), emp\_lastname (отчество), date\_hire (дата приёма на работу), emp\_percent (процент от сделки), image\_trip (необходим для предоставления изображения сотрудника), login и password (уникальный логин и пароль, необходимые для входа в систему).

Также данный класс имеет следующие операции: редатирование (Update), добавление (Insert) и удаление (Delete).

Класс «Employees» используется во всех граничных классах

Все остальные классы имеют аналогичные методы с такими же названиями.

## 1.5. Описание ER-модели базы данных

Модель сущность – связь ER-модель (entity-relationship diagram) - модель данных, на которой описаны сущности с их атрибутами и связи между ними [4].

На рисунке 8 приведена ER-диаграмма базы данных для разрабатываемой информационной системы.

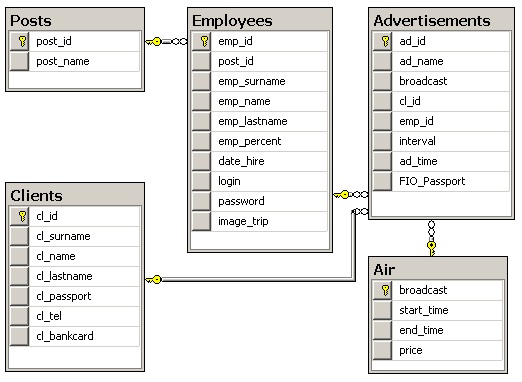


Рисунок 8 – ER-диаграмма.

Все таблицы в данной базе данных (БД) имеют связь «один ко многим» [5]:

* таблицы «Posts» и «Employees» связаны ключом «post\_id», т.к. одна и та же должность может быть у нескольких сотрудников;
* таблицы «Employees» и «Advertisements» связаны ключом «emp\_id», т.к. один сотрудник может иметь множество различных рекламных записей;
* таблицы «Clients» и «Advertisements» связаны ключом «cl\_id», т.к. один и тот же клиент может иметь несколько реклам;
* таблицы «Air» и «Advertisements» связаны ключом «broadcast», т.к. одна и та же передача может фигурировать в нескольких рекламных записях.

## ГЛАВА 2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ

В начале разработки в Visual Studio был создан проект типа Windows Forms. Приложения Windows Forms имеют интерфейс, с которым пользователь может взаимодействовать с помощью мыши и клавиатуры [6]. Для соединения приложения с базой данных использована технология Entity Framework версии 5.0.

Entity Framework представляет собой надстройку над ADO.NET.

ADO.NET осуществляет связь между БД и приложением на C#.

Двумя основными компонентами ADO.NET является поставщик данных .NET Framework и автономная модель хранения данных.

Поставщик данных .NET Framework используется для соединения с БД, выполнения команд и получение результатов посредством команд, используя язык SQL [7].

Автономная часть архитектуры, представляемая в виде класса DataSet, является кэшом данных для хранения результатов, которые получены с помощью поставщика данных.

Entity Framework является архитектурой в ADO.NET. Он позволяет обращаться к данным, используя концептуальную модель данных [8].

Entity Framework имеет ряд преимуществ: нет необходимости знать синтаксис языка запросов, вместо этого ведется работа с таблицами БД как с классами C#, где поля этих таблиц выступают в качестве свойств классов; а синтаксис SQL-запросов заменен на более удобный подход с LINQ (Language Integrated Query), а точнее LINQ to Entities [9].

LINQ to Entities предоставляет удобный подход для получения данных с помощью выражений, которые схожи выражениям языка SQL.

Entity Framework предоставляет несколько вариантов в плане проектирования БД:

* Database – First, где создается сначала БД, а затем генерируется EDMX-файл, который содержит описание EDM (Entity Data Model) в формате XML;
* Model – First – является противоположностью первого варианта, при этом изначально в дизайнере создается описание EDM, затем на её основе генерируется БД;
* Code - First – с помощью классов C# создаётся БД.

При создании приложения в данном курсовом проекте был использован вариант Database – First [10].

## 2.1. Подключение к базе данных

Для начала необходимо создать концептуальную модель и контекст (рисунок 9) на основе существующей БД. Для этого нужно:

1. выбрать в меню элемент «Проект»;
2. выбрать в этом меню «Добавить новый элемент»;
3. выбрать «Модель ADO.NET EDM»;
4. выбрать вариант «Создать из базы данных»;
5. создать соединение с БД;
6. присвоить названию контекста значение AdvertisingEntities;
7. выбрать все объекты, которые необходимо включить в модель – в данном проекте были использованы пять таблиц и одно представление

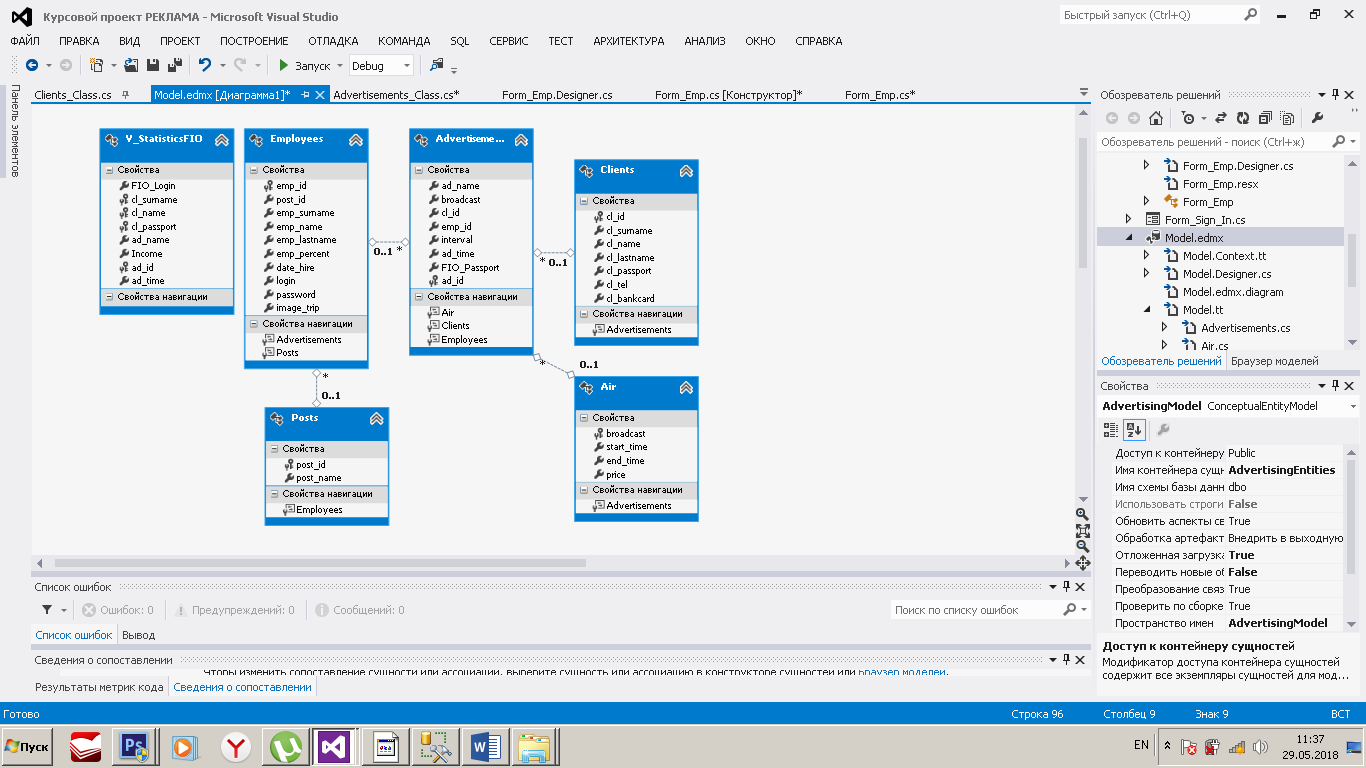


Рисунок 9 – EDM-модель.

## 2.2. Вход в систему

При запуске приложения пользователю предоставляется форма «Авторизация» для входа в «Главное меню» (рисунок 10), где ему необходимо указать свои логин и пароль. Также на форме «Главное меню» имеется кнопка для выхода из системы.

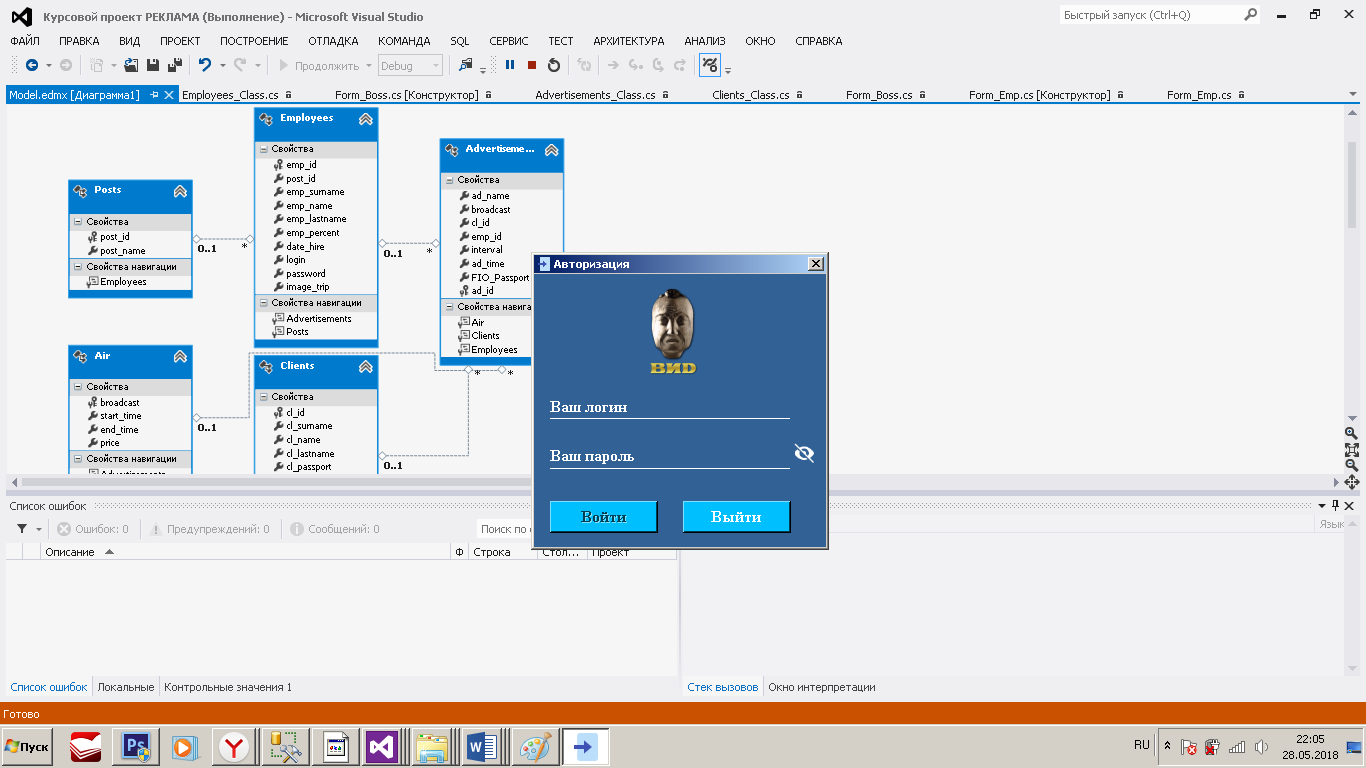


Рисунок 10 - форма «Авторизация»

В зависимости от введённых пользователем данных, открывается либо форма для главы отдела, либо для рекламного агента. Ниже представлен код, который предоставляет возможность перехода к «Главному меню» (рисунок 11).

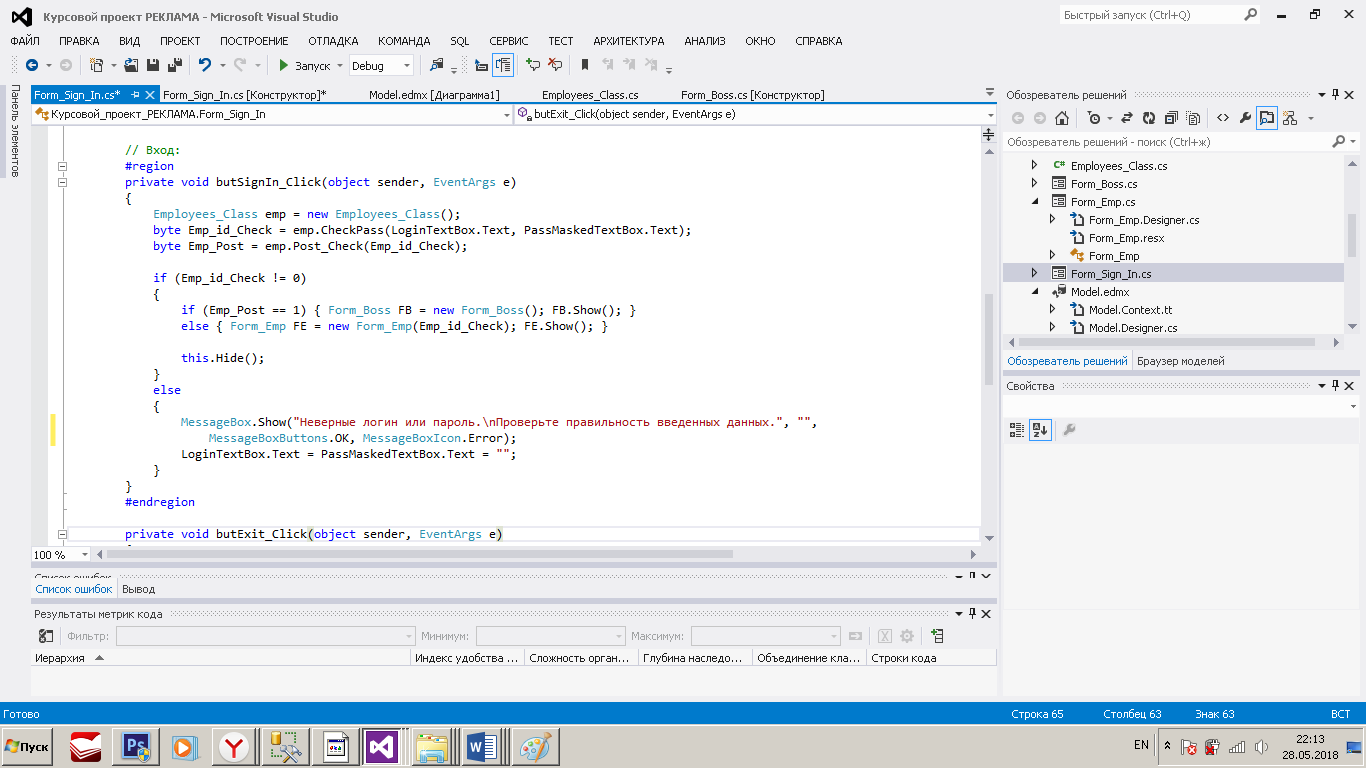


Рисунок 11 – код для перехода в «Главное меню».

## 2.3. Ведение информации о клиенте

Рассмотрим вариант, когда сотрудник вошёл в систему.

Ему предоставляется форма (рисунок 12), где расположены все необходимые для его деятельности функции.

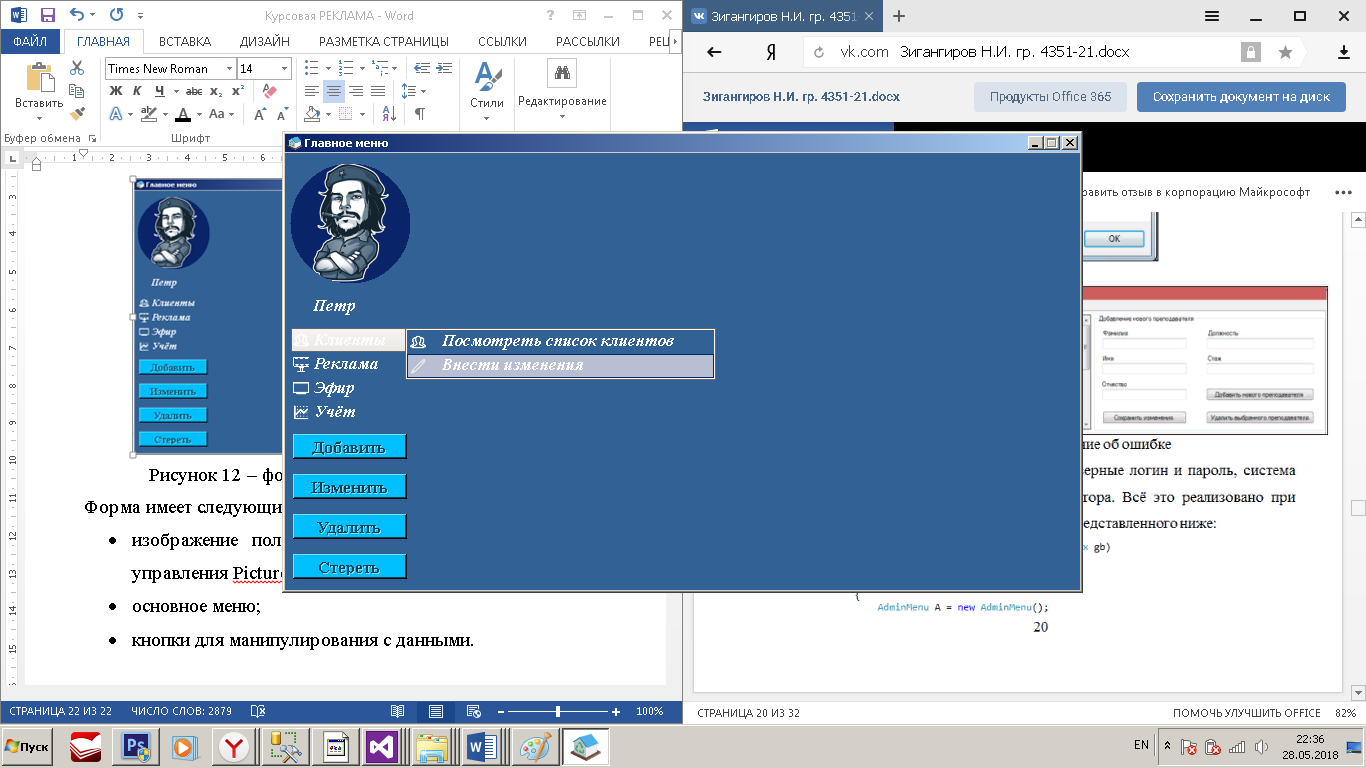


Рисунок 12 – форма «Главное меню» для сотрудника.

Форма имеет следующие объекты:

* изображение пользователя, которое размещено в элементе управления PictureBox;
* основное меню;
* кнопки для манипулирования с данными.

Меню «Клиенты» содержит подменю с двумя элементами:

* посмотреть список клиентов;
* внести изменения.

При выборе первого элемента загружается dataGridView, который содержит в себе информацию о клиентах, хранимых в базе (рисунок 13).

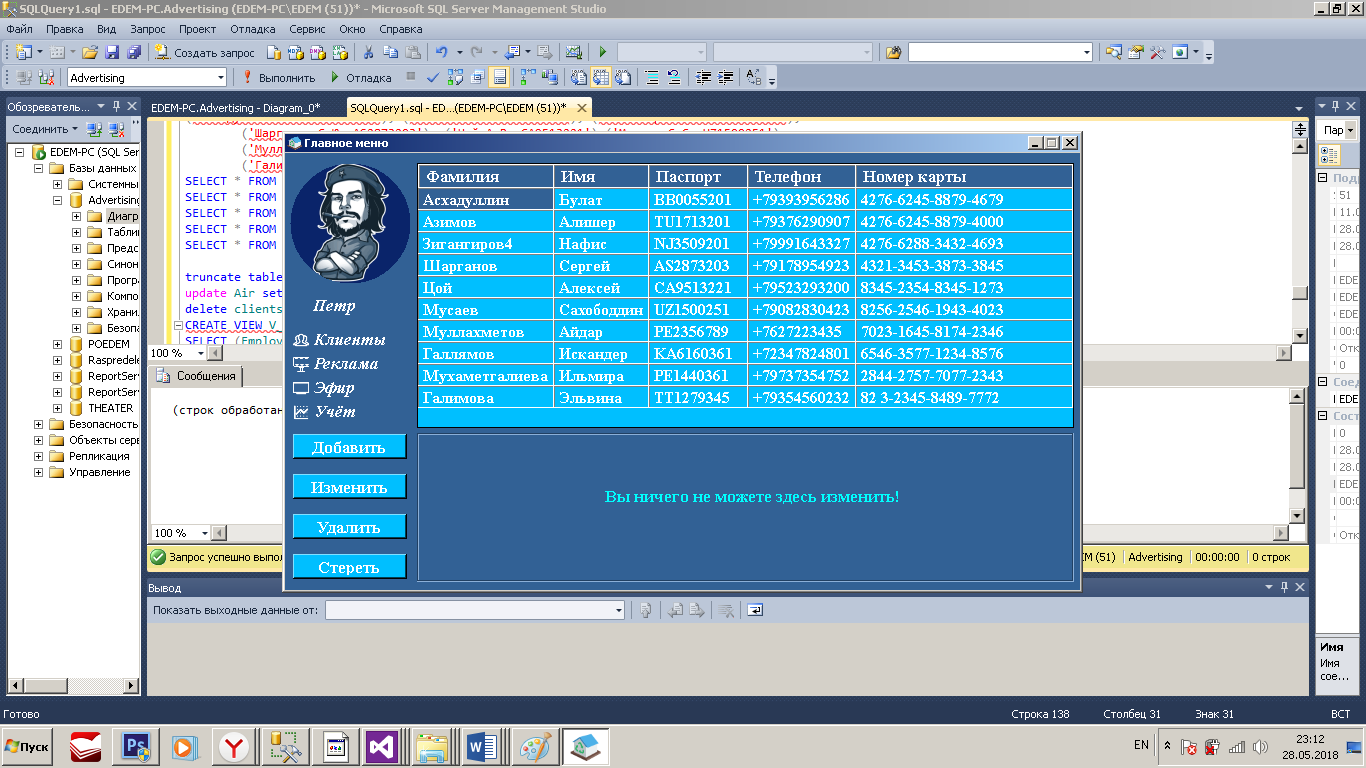


Рисунок 13 – Список клиентов.

При выборе второго варианта (рисунок 14) открывается панель для ввода данных и dataGridView с клиентами, которые были добавлены пользователем. При двойном нажатии по записи в таблице, панель автоматически заполняется соответствующими данными. После чего пользователю доступно две функции: удалить клиента или изменить его данные. Код для каждой функции представлен на рисунках 15-18.

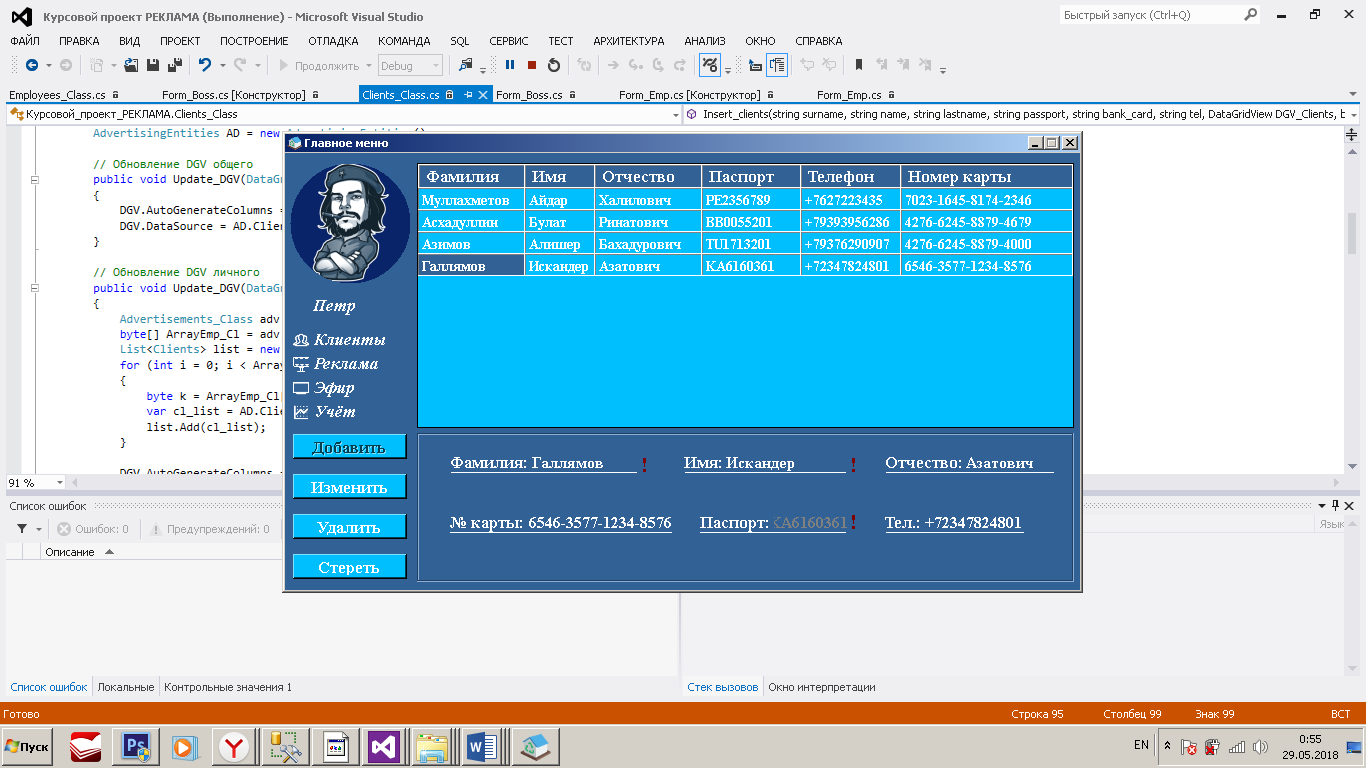


Рисунок 14 – форма для изменений данных

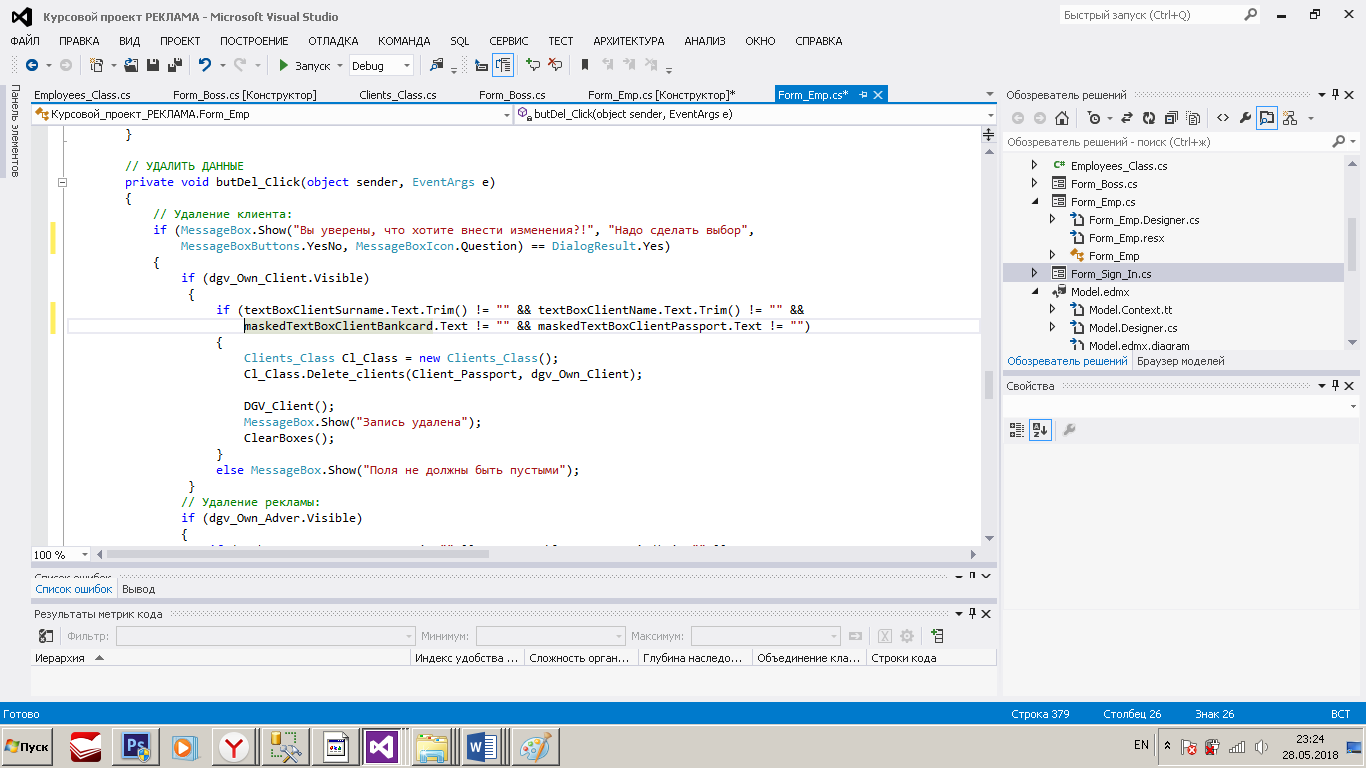


Рисунок 15 – код с вызовом метода для удаления записи в обработчике события кнопки удаления в «Главном меню».

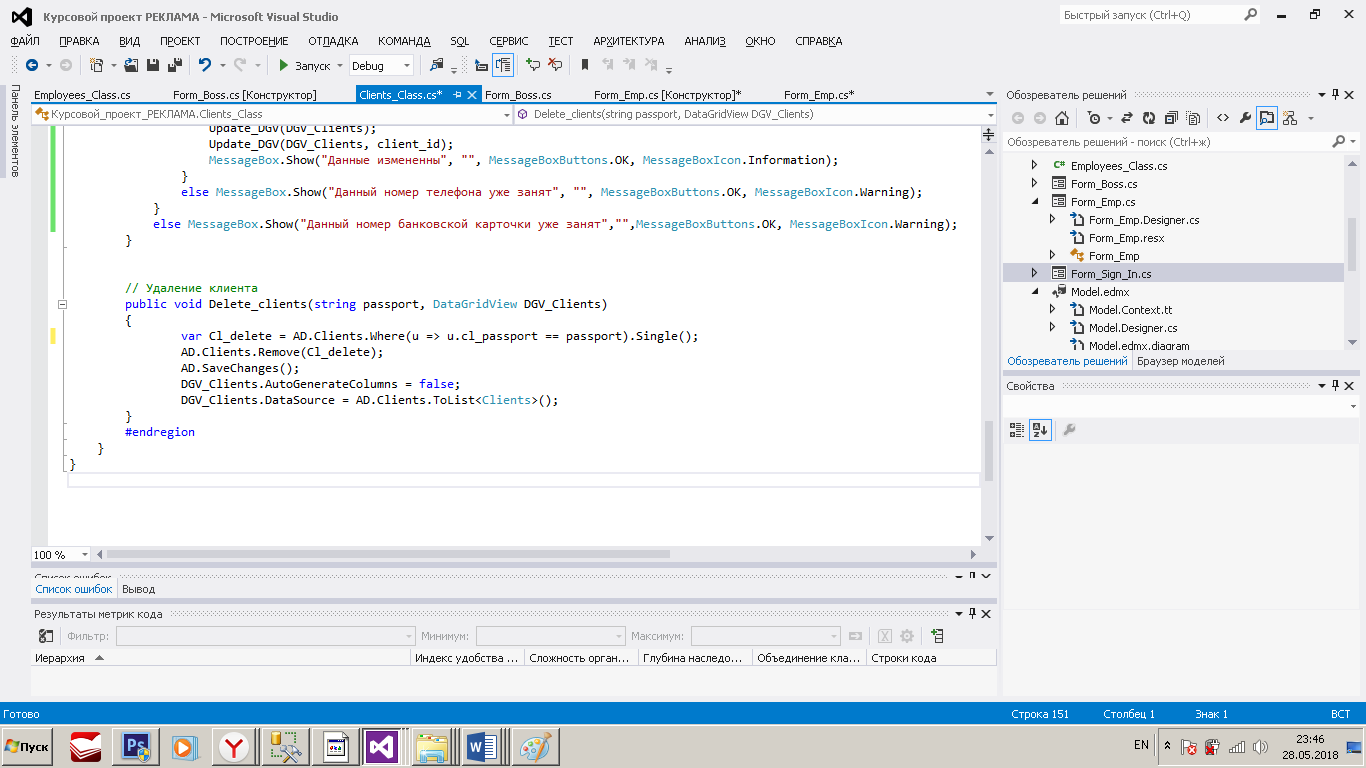


Рисунок 16 – код для удаления в методе, реализованного в классе.

В данном примере вызывается метод «Delete\_clients», находящийся в соответствующим классе. В качестве параметров передаются паспортные данные клиента, которого необходимо удалить и dataGridView для обновления таблицы.

В методе «Delete\_clients», создаётся нетипизированная переменная «Cl\_delete», которой присваивается значение записи, где паспортные данные совпадают с входным параметром «passport». Для реализации предыдущего шага было использовано лямбда – выражение (лямбда – выражения представляют собой упрощенную запись кода, позволяющие создать ёмкие и лаконичные методы).

После присвоения переменной «Cl\_delete» некоторого значения, реализуется встроенный метод «Remove» для удаления из базы данных выбранной записи, сохраняя все изменения в базе. Далее, используя в качестве источника данных (DataSource) сущность «Clients», dataGridView заполняется обновленными данными.

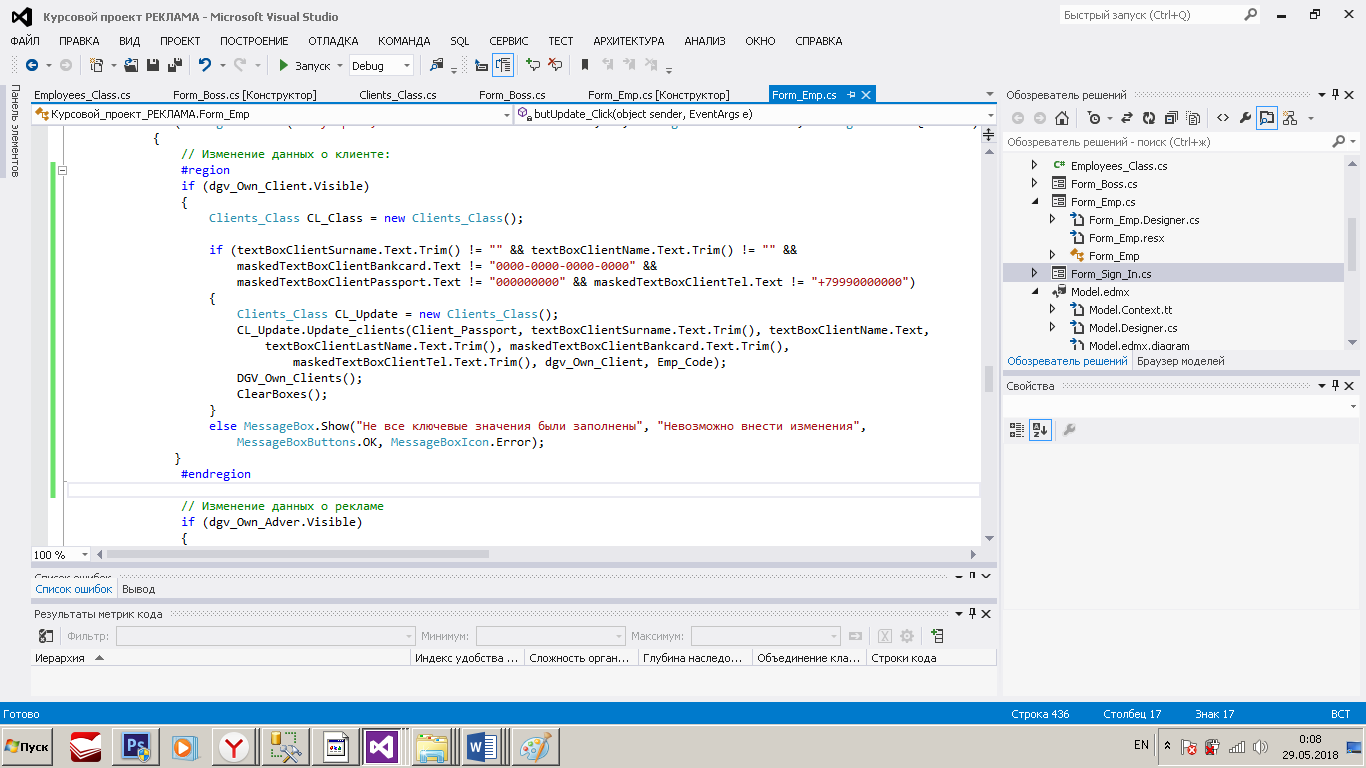


Рисунок 17 – код с вызовом метода для обновления записи в обработчике события кнопки удаления в «Главном меню».

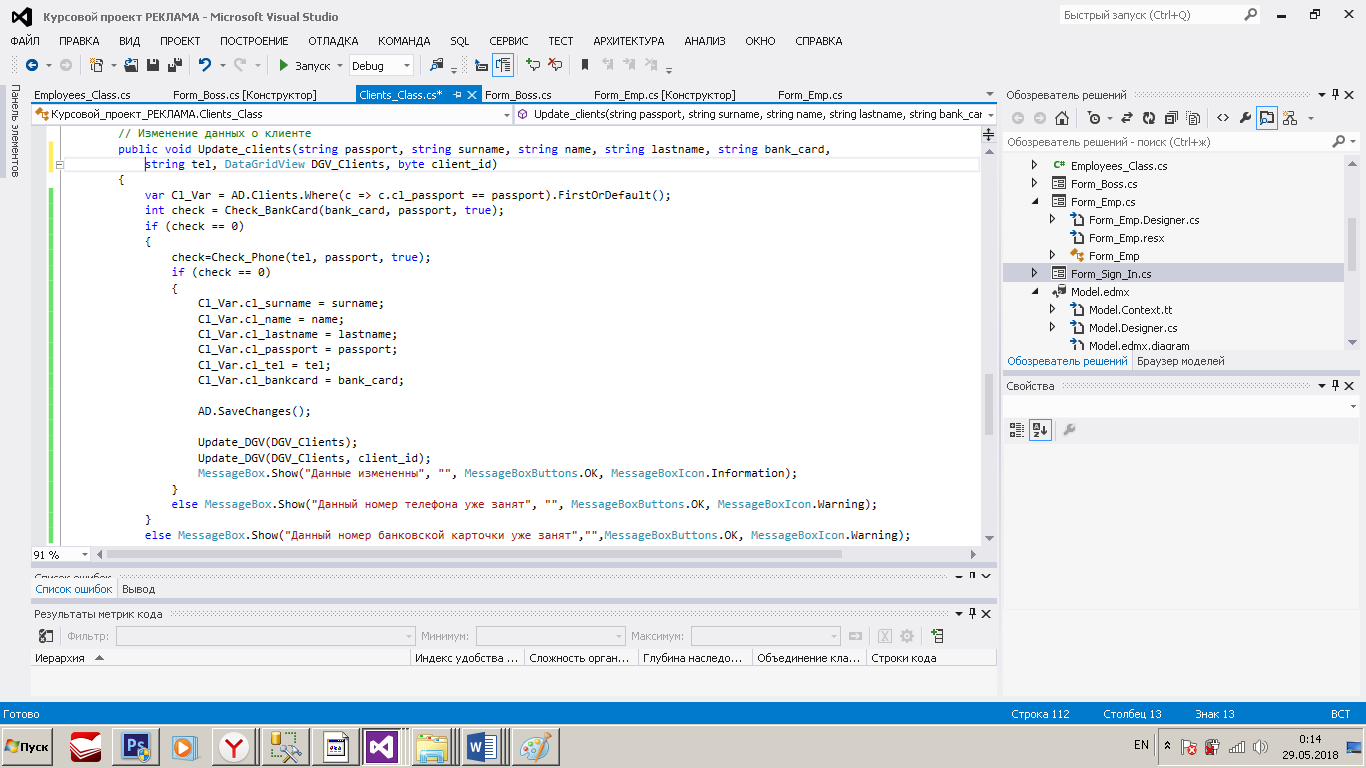


Рисунок 18 – код для изменения клиента в методе, реализованного в классе.

В обработчике событий для кнопки обновления вызывается метод «Update\_clients», находящийся вcё в том же классе. В качестве параметров передаются паспортные данные клиента, которого необходимо изменить и dataGridView для обновления таблицы.

В методе «Update\_clients», создаётся нетипизированная переменная «Cl\_Var», которой присваивается значение записи, где паспортные данные совпадают с входным параметром «passport».

Также происходит проверка на наличие таких же банковских реквизитов или номера телефона (рисунок 19). Если такие данные уже имеются в базе, тогда выводится соответствующее сообщение и изменения отменяются. Если же совпадений не оказалось, то свойства переменной «Cl\_Var» принимают новые значения, а именно значения параметров, например, свойство cl\_bankcard становится равным параметру bank\_card. Затем все изменения сохраняются и идёт обновление dataGridView с помощью метода «Update\_DGV()» (рисунок 20).

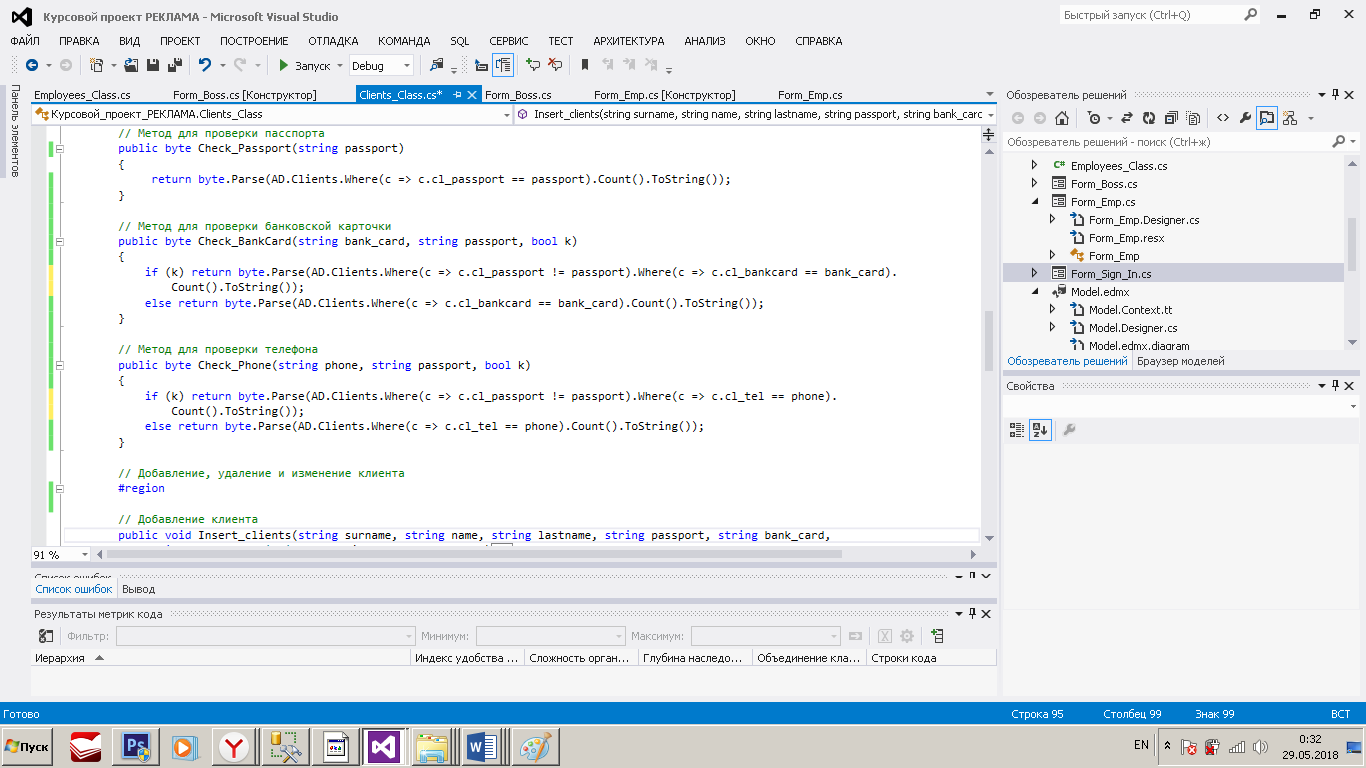


Рисунок 19 – методы для проверки банковской карточки и телефона

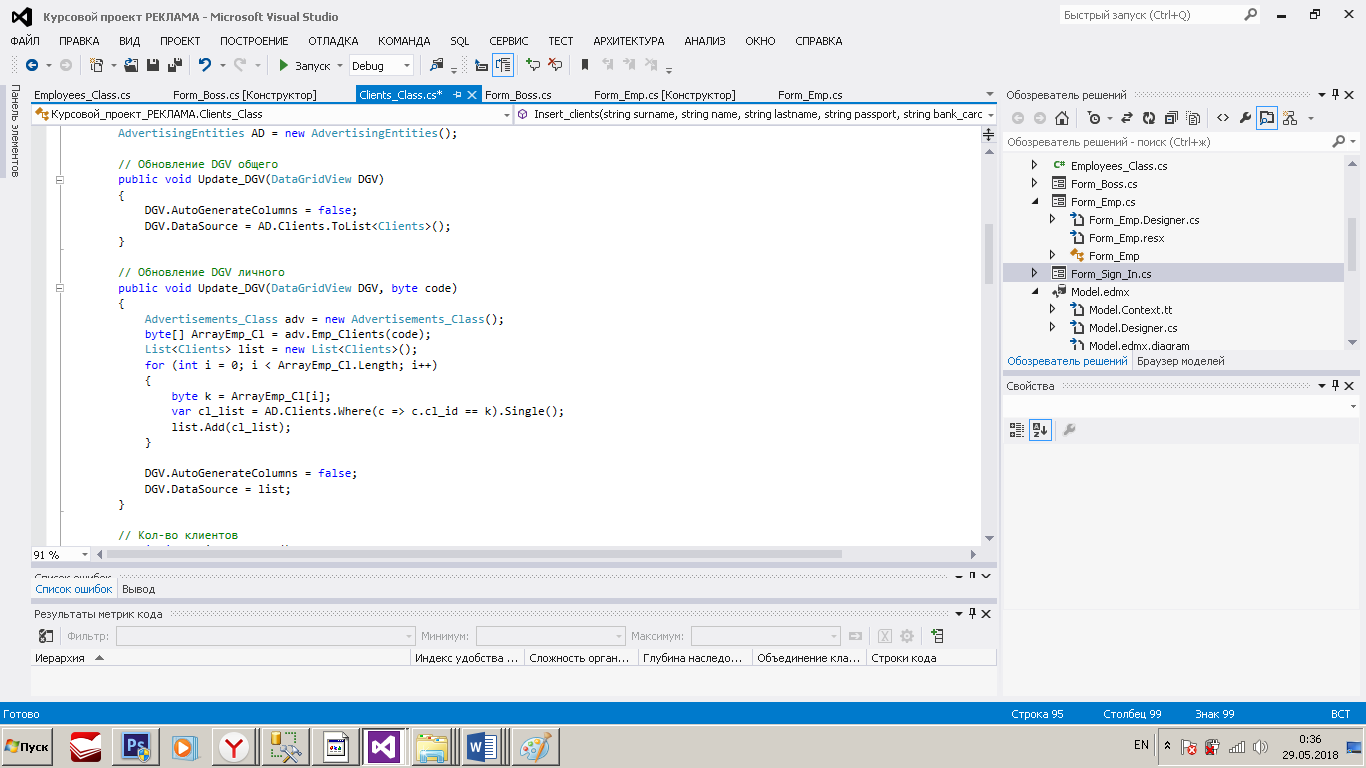


Рисунок 20 – методы для обновления dataGridView

Для того, чтобы добавить нового клиента, необходимо заполнить все поля (рисунок 21) самостоятельно. После чего пользователю доступна функция добавления.

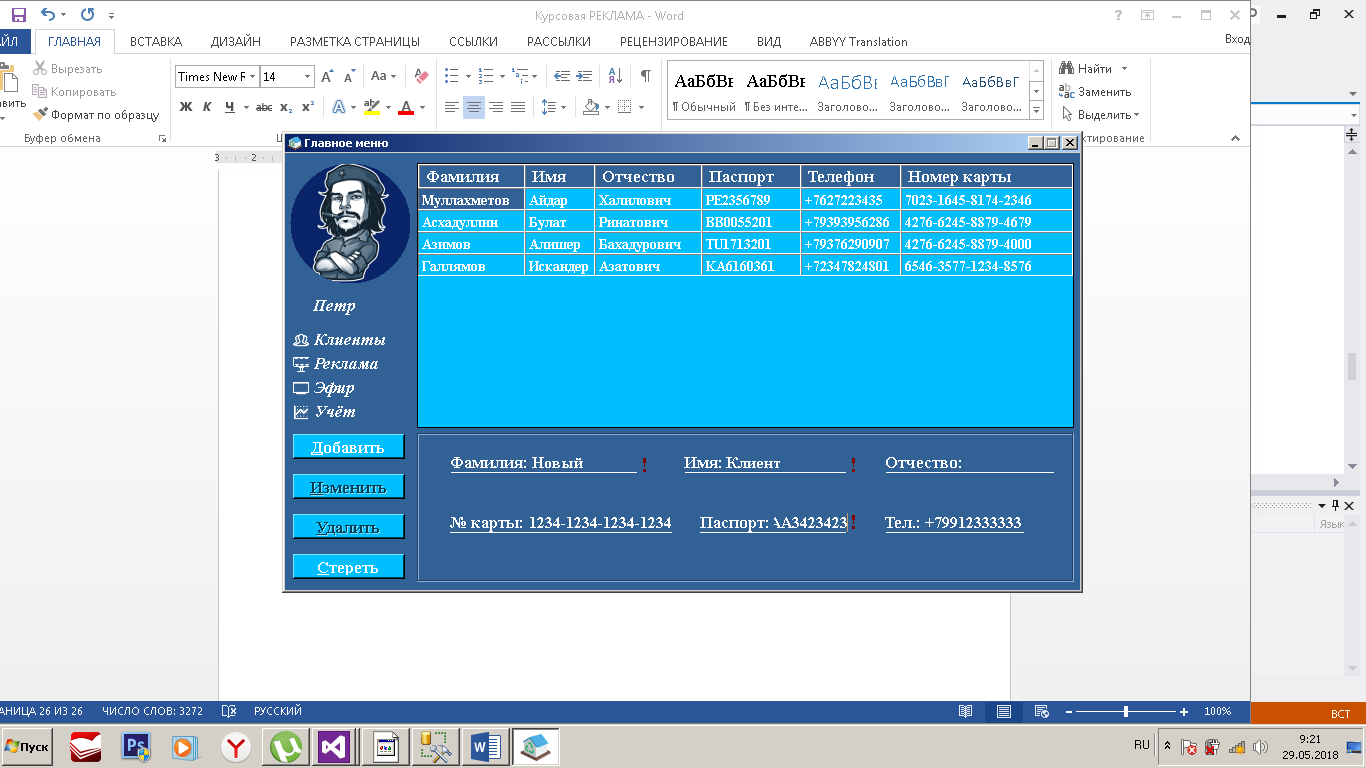


Рисунок 21 – форма с функцией добавления клиента

После добавления клиента в базу, выводится следующее сообщение (рисунок 22):

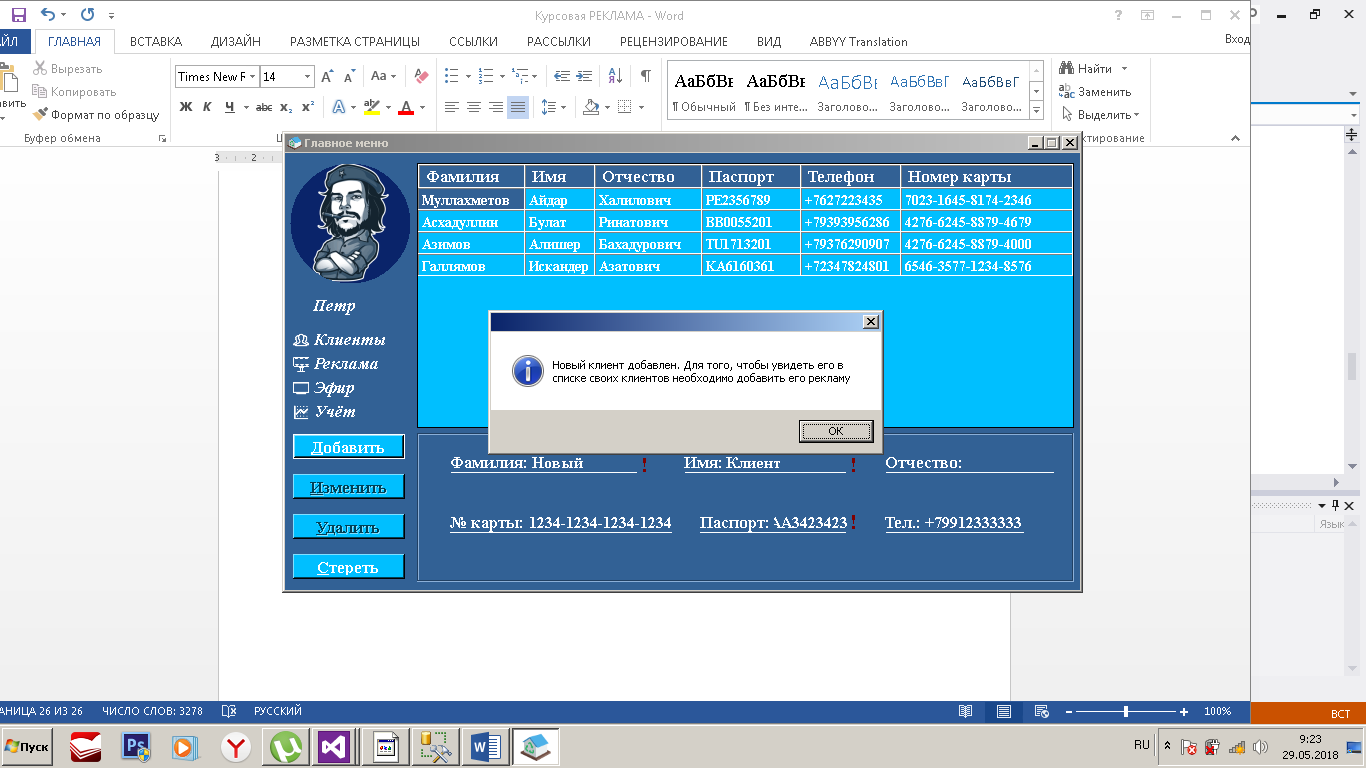


Рисунок 22 – сообщение после успешного добавления клиента.

Ниже на рисунках 23-24 представлен код для добавления клиента.

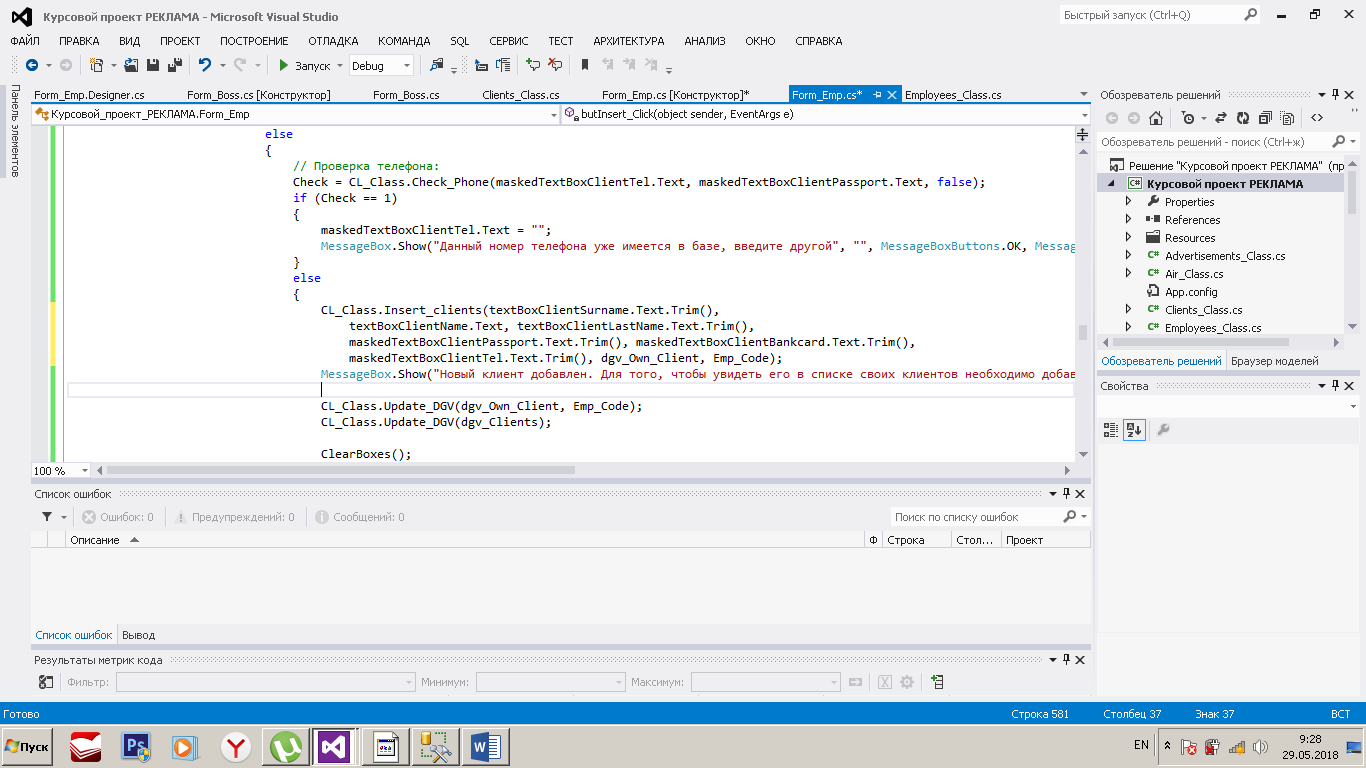


Рисунок 23 – вызов метода для добавления клиента в обработчике события кнопки «Добавить».

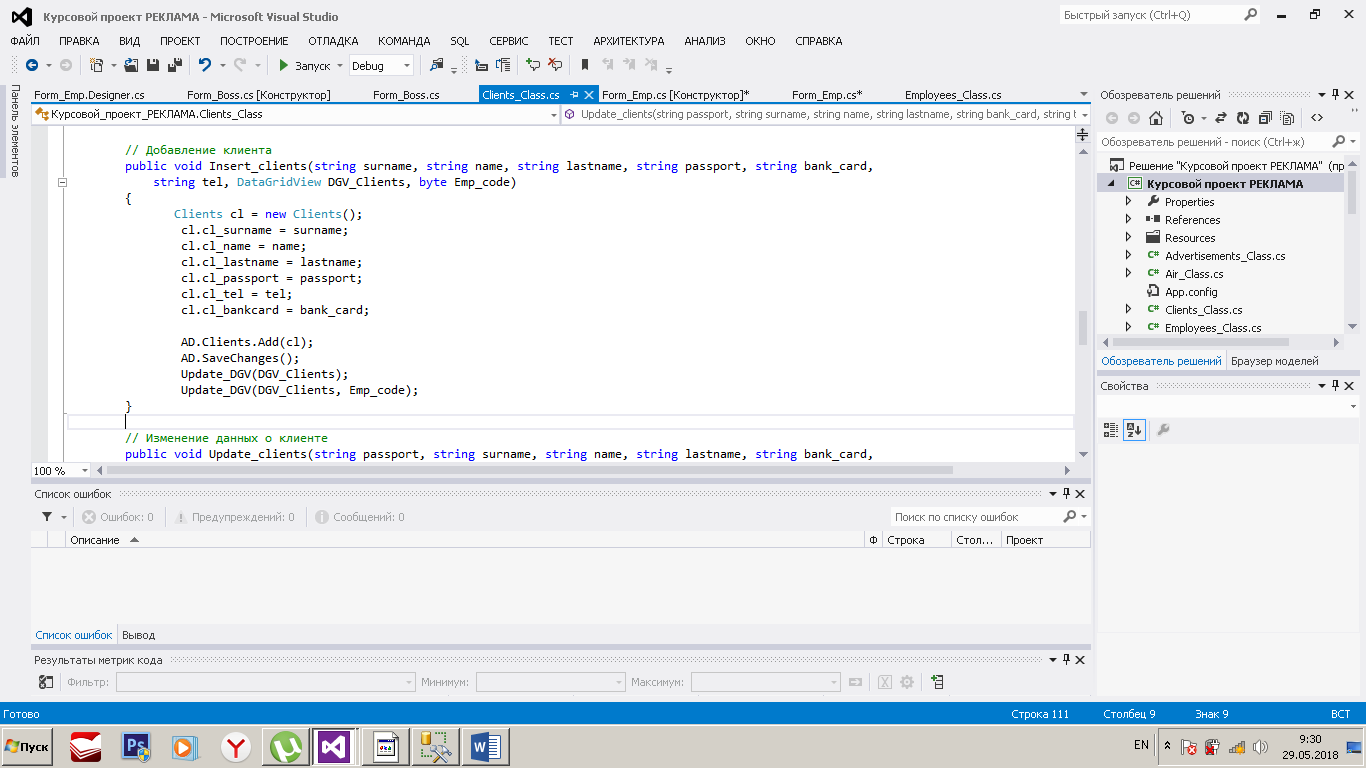


Рисунок 24 – метод добавления клиента, реализованный в классе.

Добавление клиента после заполнения всех необходимых данных осуществляется следующим образом:

1. вызывается метод, который принимает в качестве параметров заполненные поля на форме, также логин и пароль, сгенерированные случайным образом;
2. непосредственно в методе создаётся экземпляр класса и его свойствам присваиваются значения входных параметров
3. далее происходит добавление экземпляра класса в контекст с помощью метода Add();
4. затем следует сохранение изменений с помощью метода SaveChanges().

Для того, чтобы увидеть данные нового клиента следует выбрать пункт «Посмотреть список клиентов» в меню «Клиенты».

Теперь добавленный клиент доступен для оформления рекламы (рисунок 25).

## 2.4. Ведение информации о рекламе

Добавим новую рекламу на только что созданного клиента. Для этого необходимо выбрать пункт «Внести изменения» в меню «Реклама». Внести все необходимые данные и нажать на кнопку «Добавить» (рисунок 26).

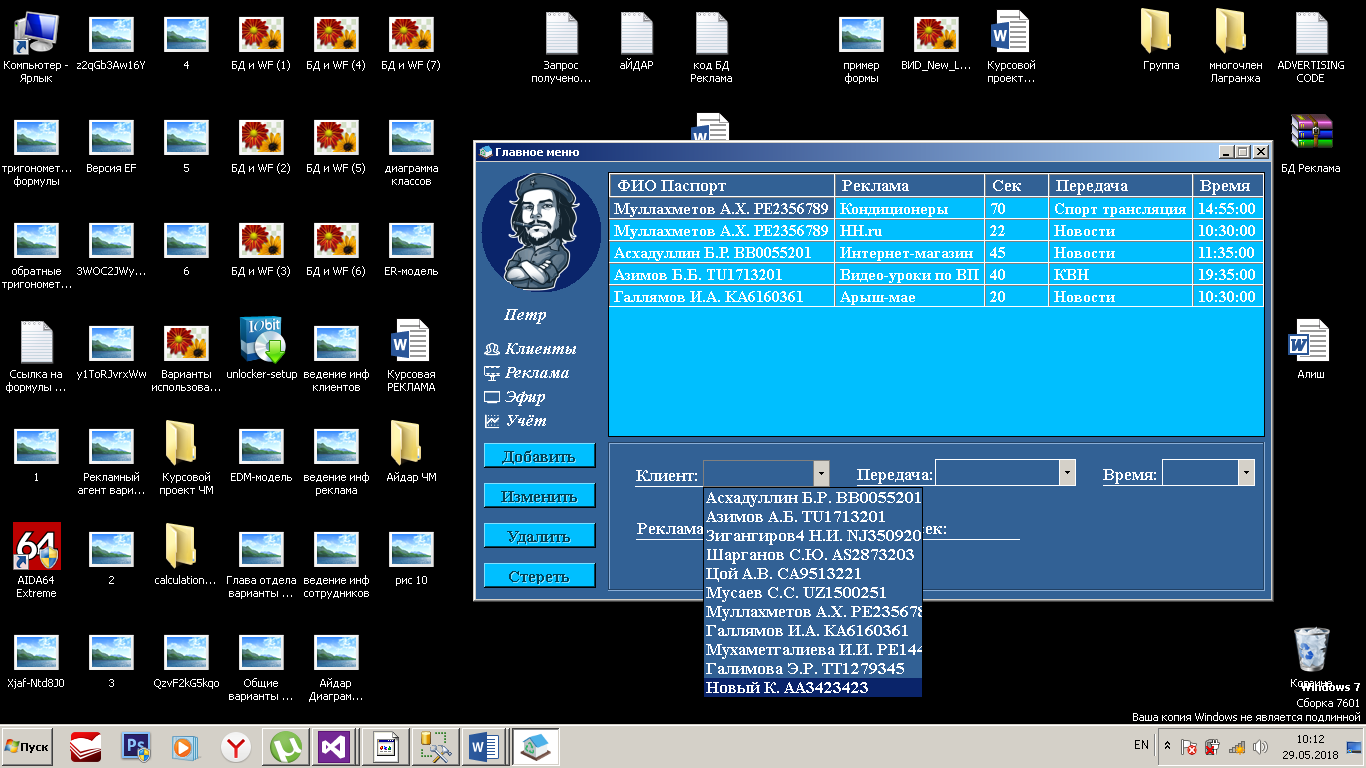


Рисунок 25 – выбор нового клиента для оформления рекламы.

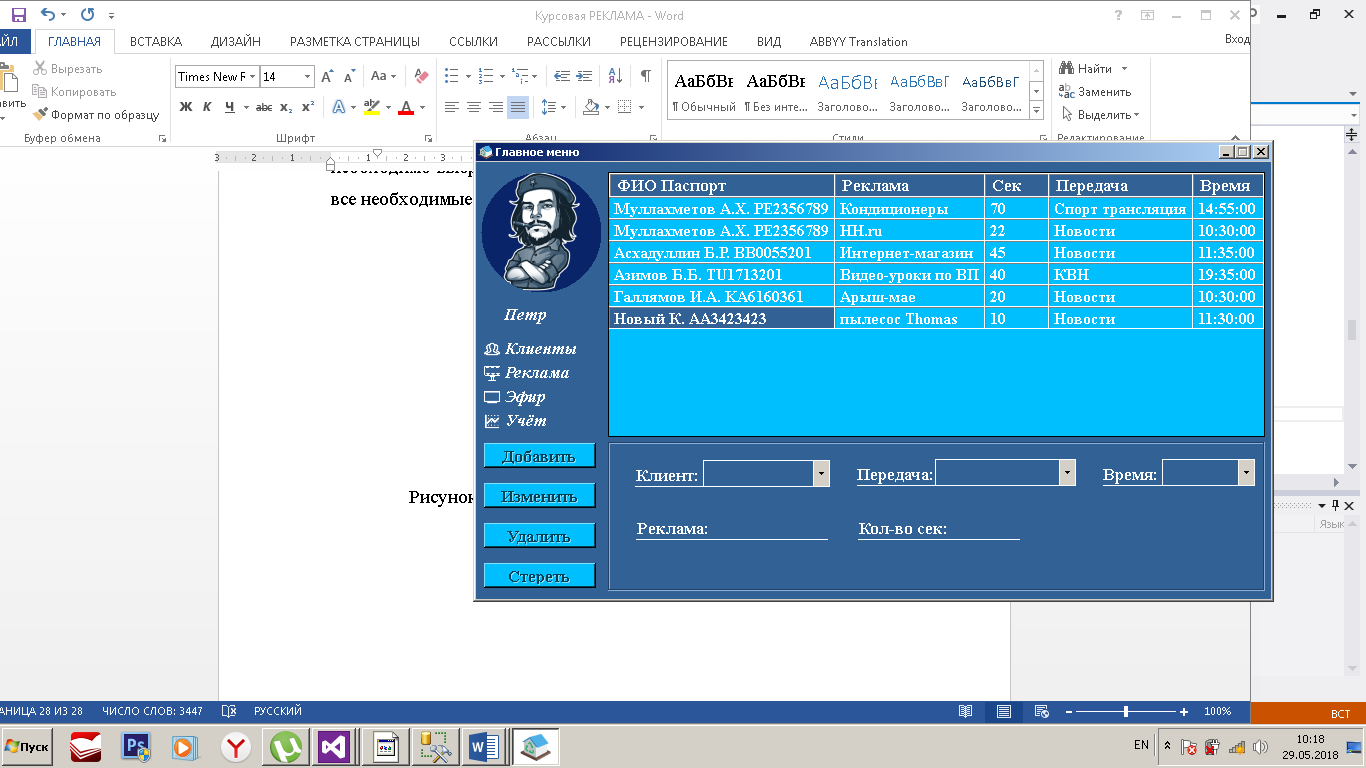


Рисунок 26 – результат добавления новой рекламы.

На рисунках 27-28 изображён код для функции добавления рекламы.

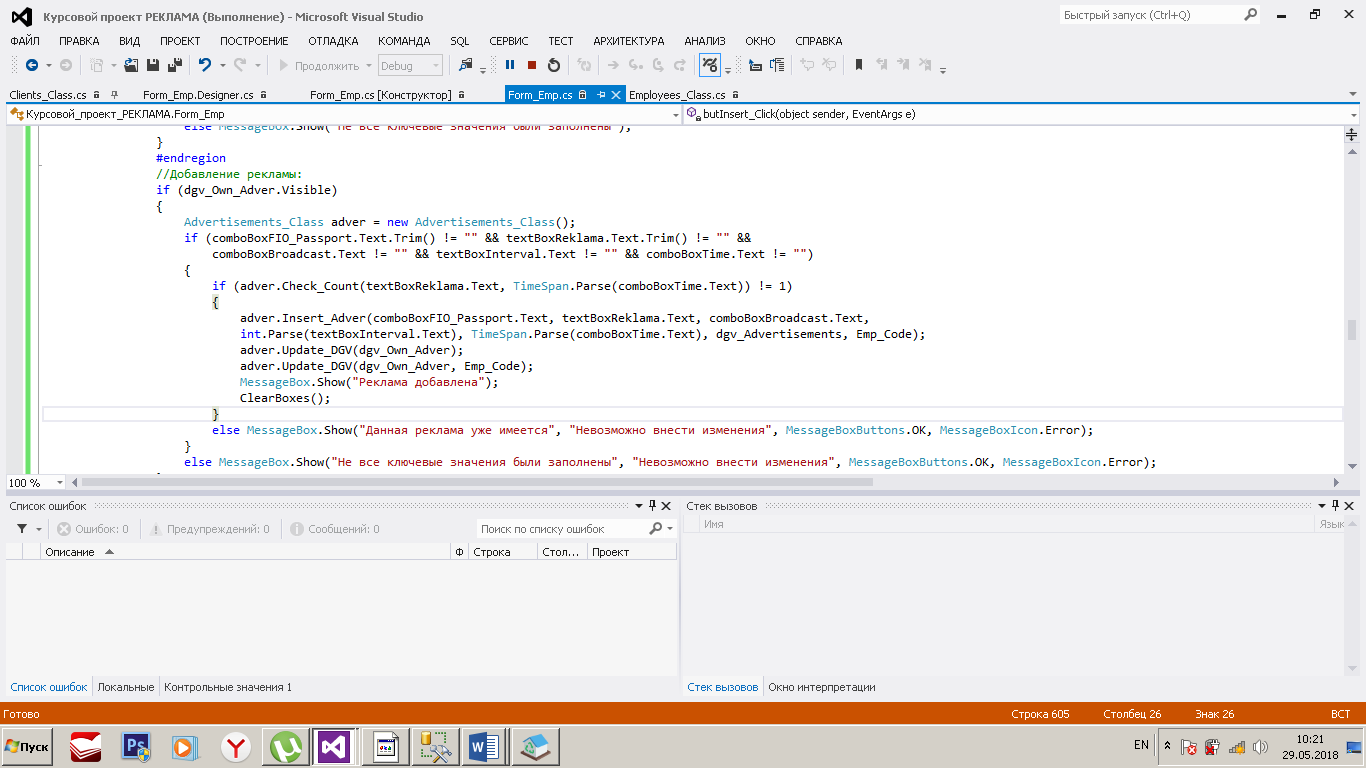


Рисунок 27 – вызов метода для добавления рекламы в обработчике события кнопки «Добавить».

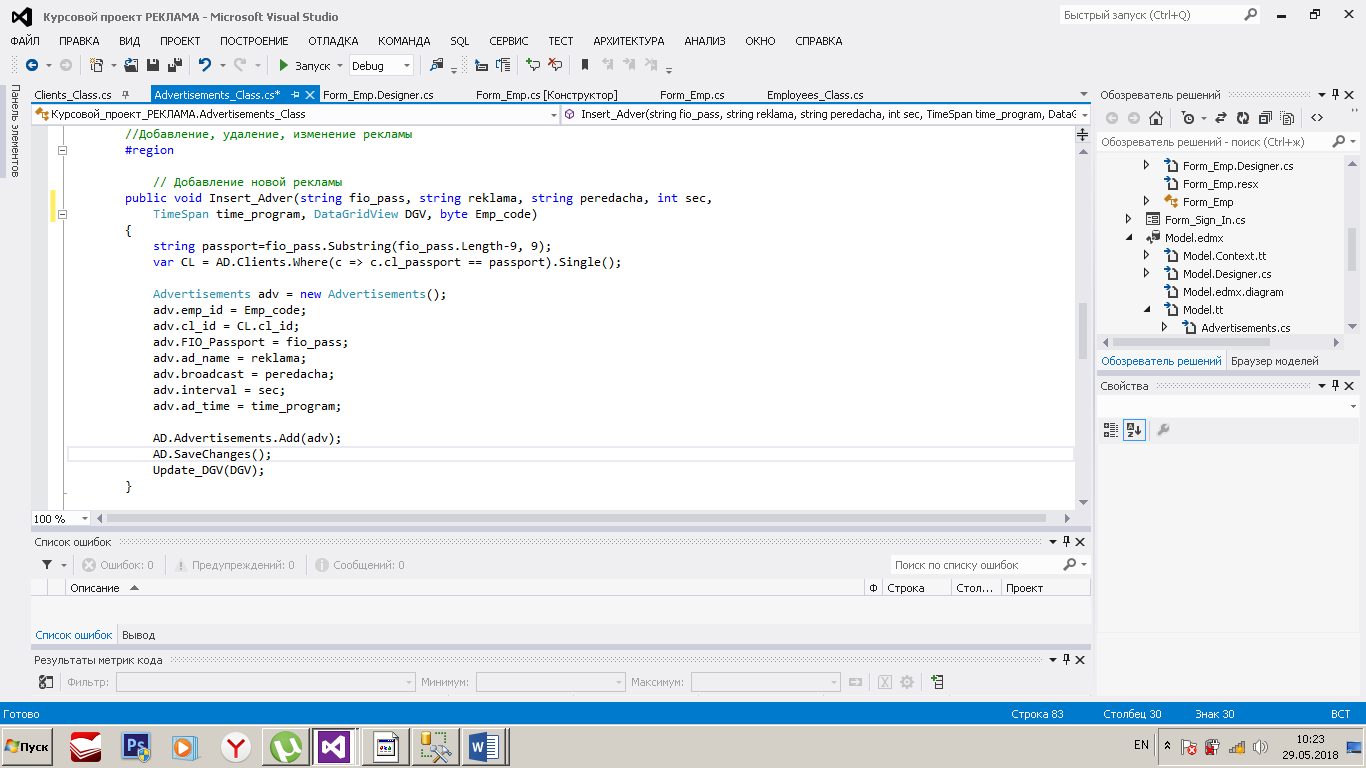


Рисунок 28 – метод добавления рекламы, реализованный в классе.

Добавление рекламы имеет такой же алгоритм, как и добавление клиента.

Далее рассмотрим редактирование данных о рекламе и её удаление из базы.

Для того, чтобы изменить информацию о рекламе, например, поменять время показа и её длительность, необходимо дважды щелкнуть по определённой записи в таблице. После этого данные автоматически переходят на форму для заполнения. Редактируем определённые данные и нажимаем на кнопку «Изменить» (рисунок 29). На рисунках 30 – 31 представлен код для редактирования рекламы.



Рисунок 29 – результат изменения данных о рекламе.

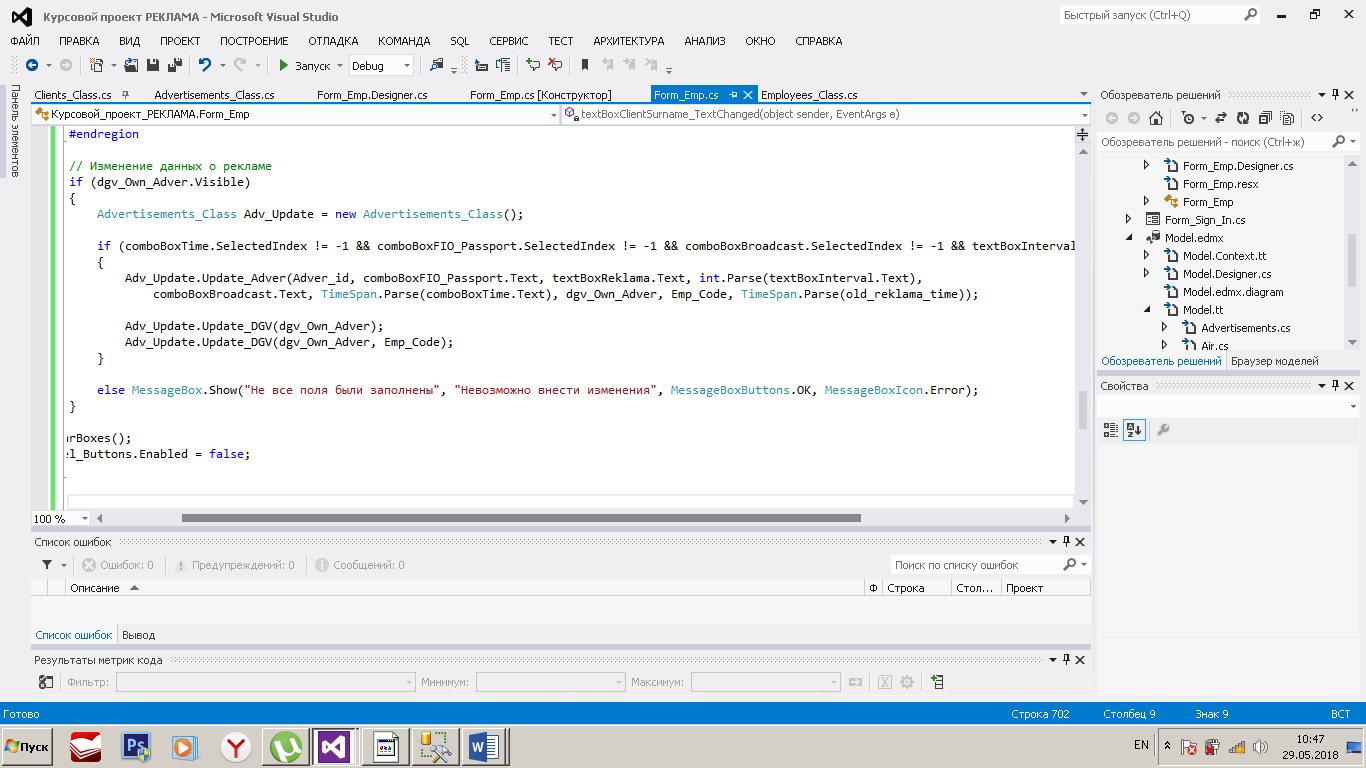


Рисунок 30 – вызов метода для изменения данных о рекламе в обработчике события кнопки «Изменить».

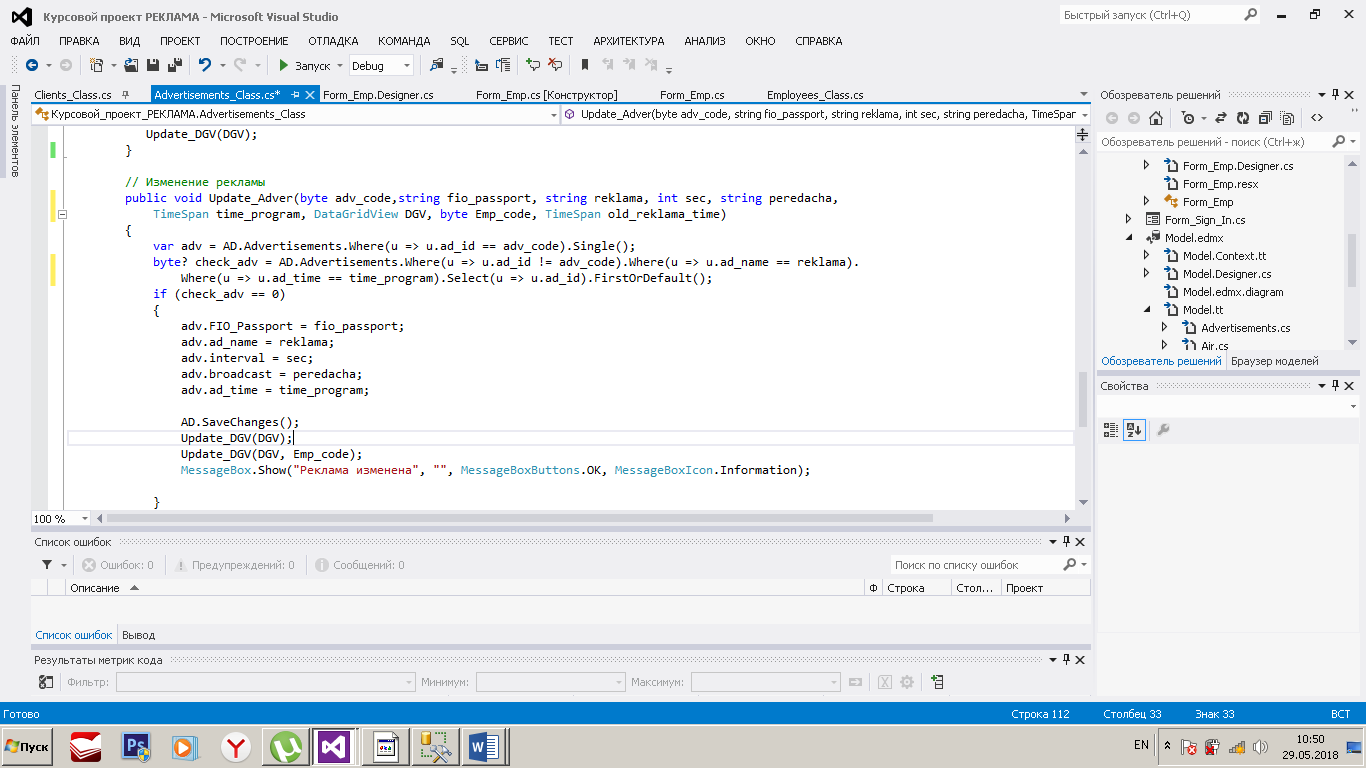


Рисунок 31 – метод редактирования рекламы, реализованный в классе.

Алгоритм изменения рекламы аналогичен алгоритму редактирования данных о клиенте. На рисунке 31 также предусмотрена проверка на наличие рекламы с таким же названием и временем показа.

Для того, чтобы удалить определённую рекламу из базы, необходимо дважды щелкнуть по ней в таблице. После этого нажимаем на кнопку «Удалить» (рисунок 32). На рисунках 33 – 34 представлен код для удаления рекламы.

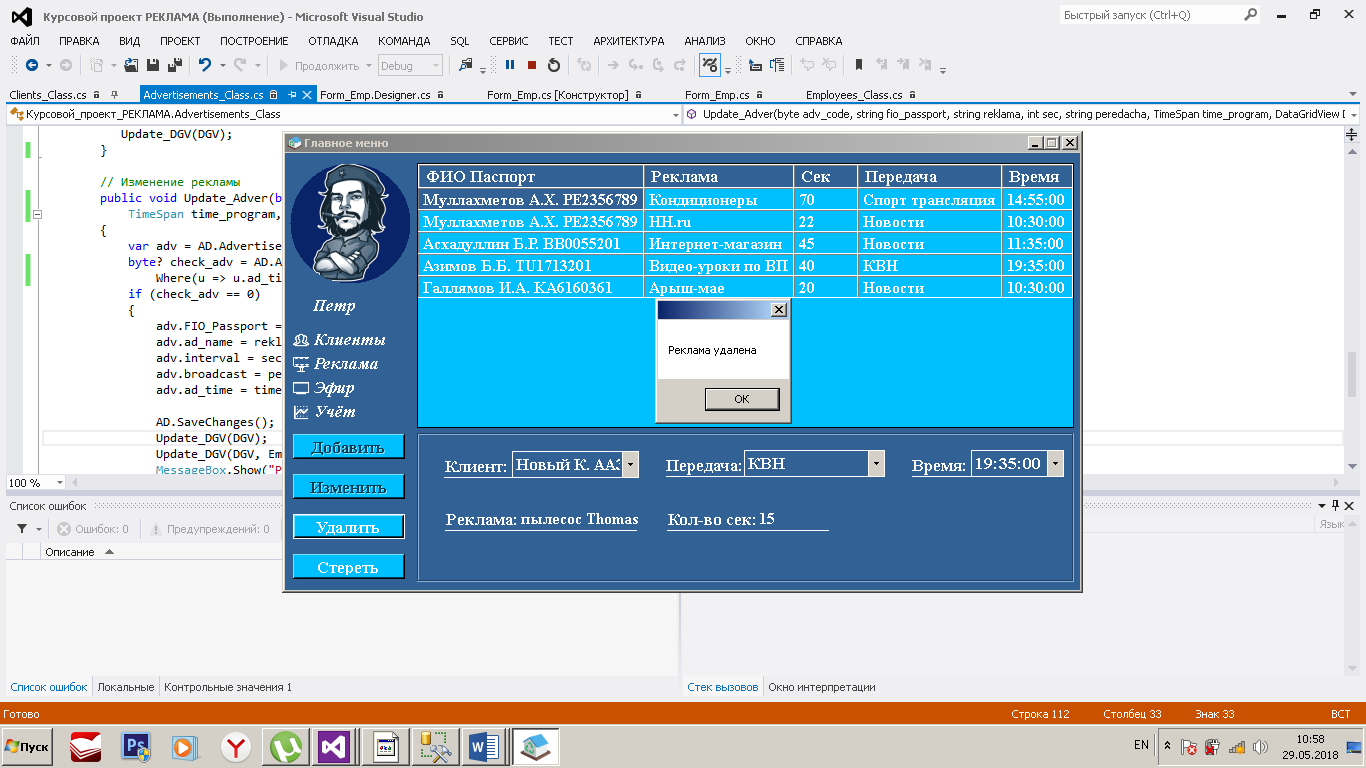


Рисунок 32 – результат удаления рекламы.

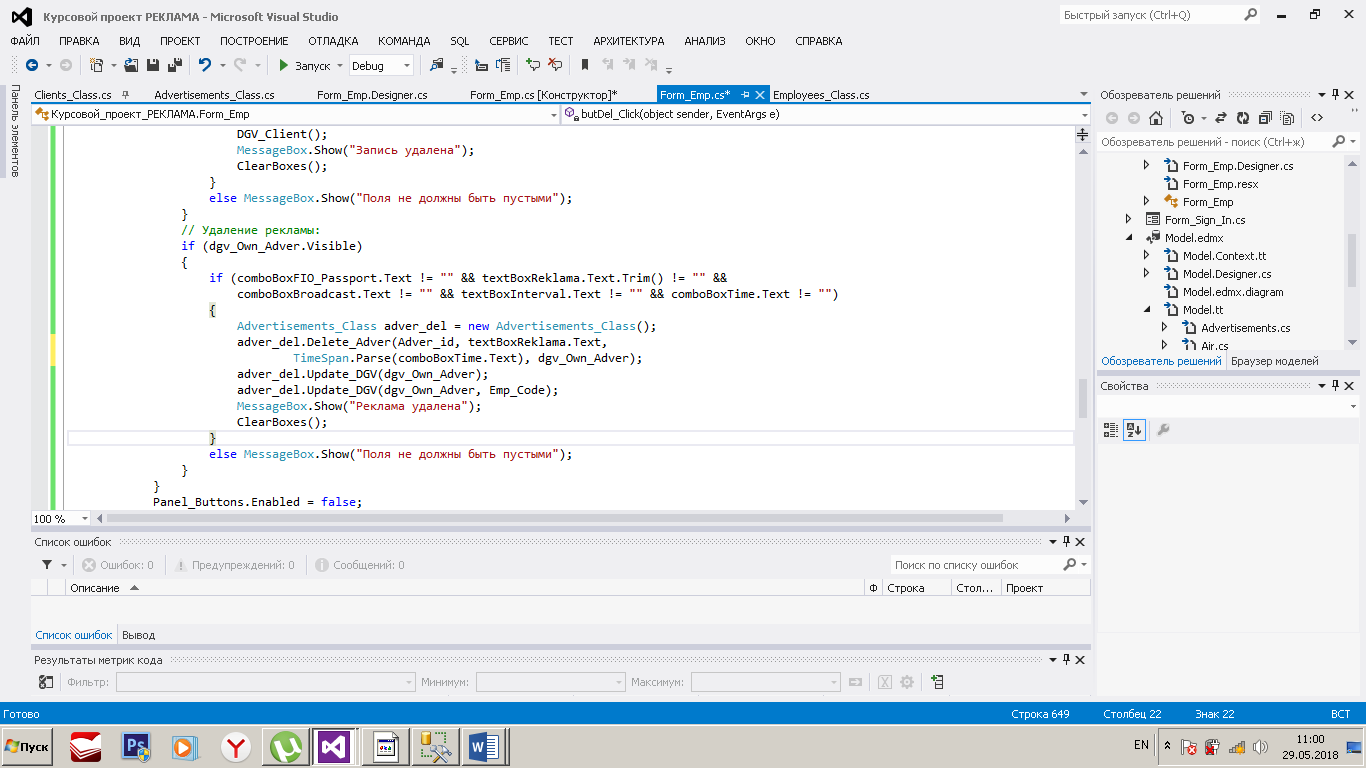


Рисунок 33 – вызов метода для удаления рекламы в обработчике события кнопки «Удалить».

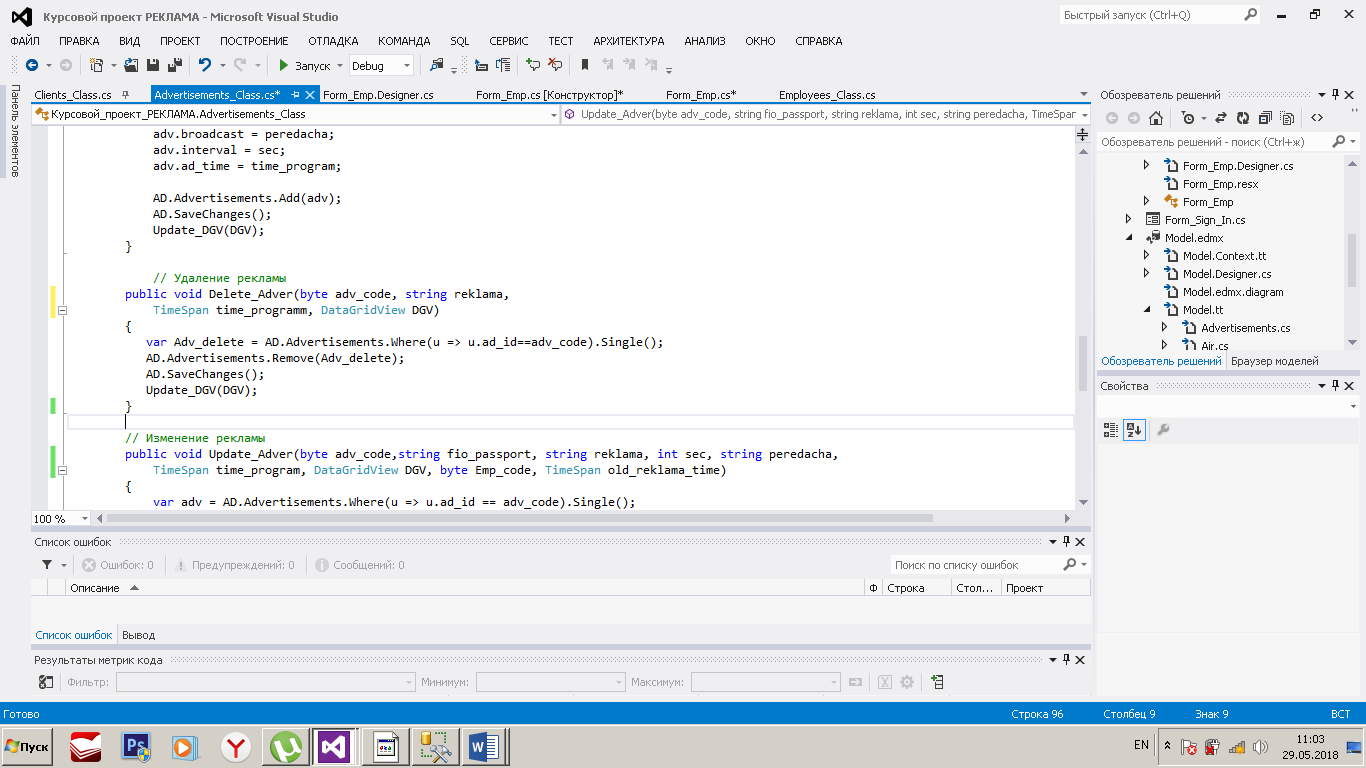


Рисунок 34 – метод удаления рекламы, реализованный в классе.

Алгоритм удаления рекламы аналогичен алгоритму удаления клиента.

## 2.5. Просмотр личной статистики

Для того, чтобы посмотреть на личную статистику пользователя необходимо выбрать меню «Учёт» (рисунок 35).

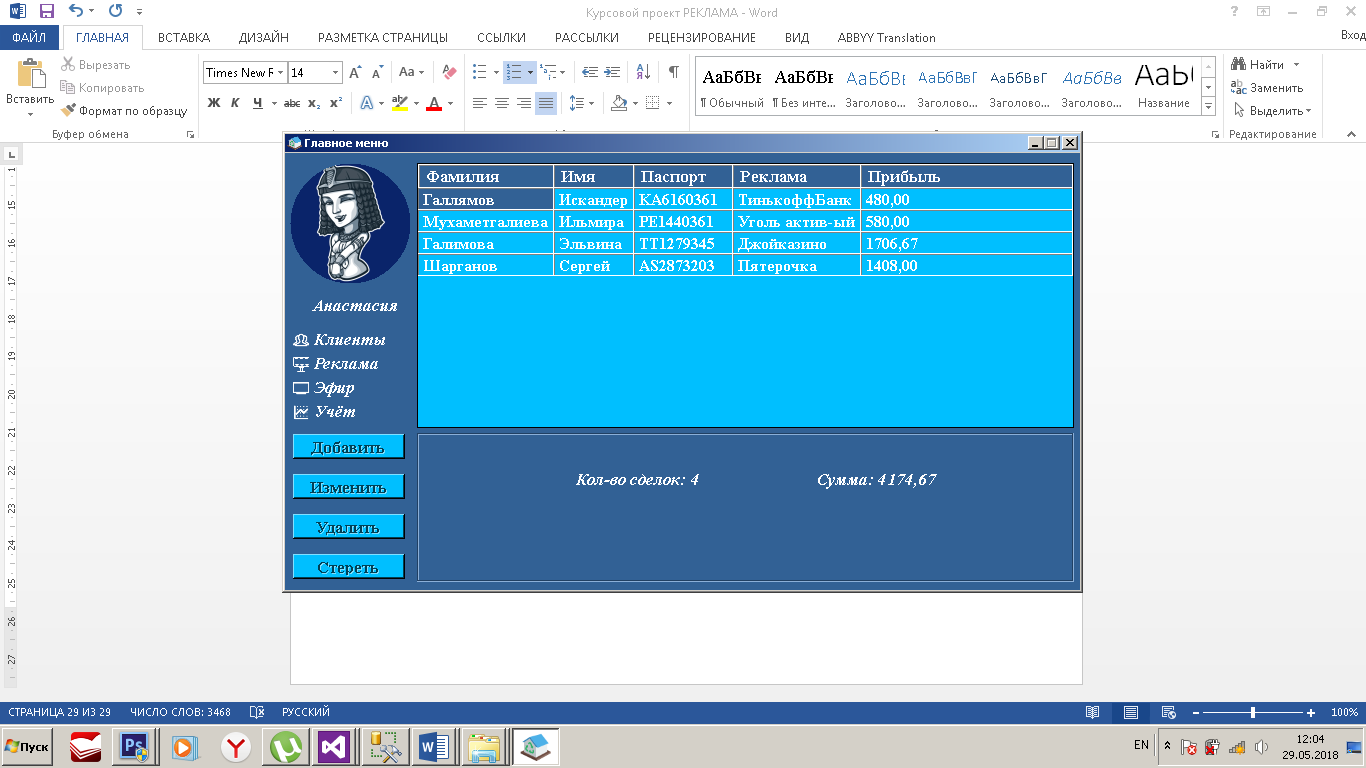


Рисунок 35 – личная статистика.

На рисунке 36 - 37 представлен код для реализации данной функции.

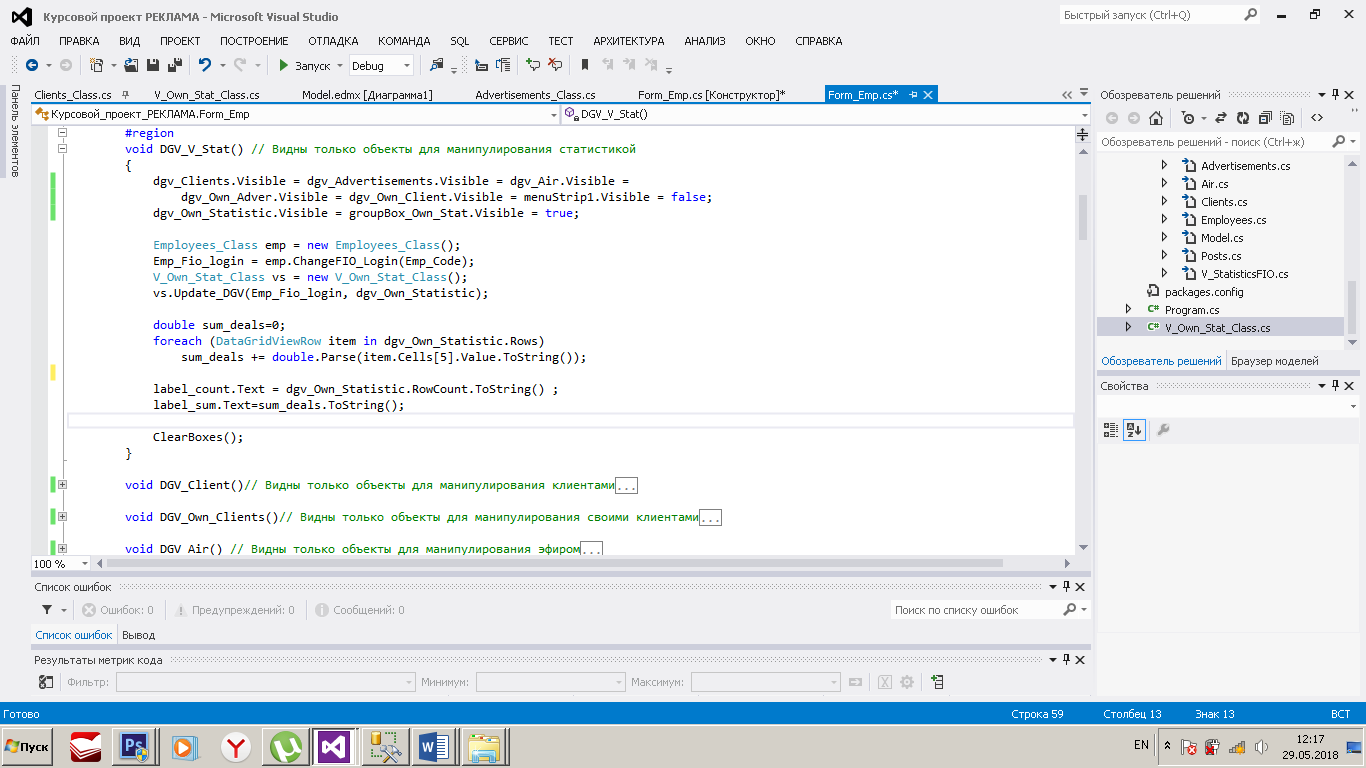


Рисунок 36 – вызов метода для вывода всех сделок сотрудника и код для вычислений, реализованные в «Главном меню»

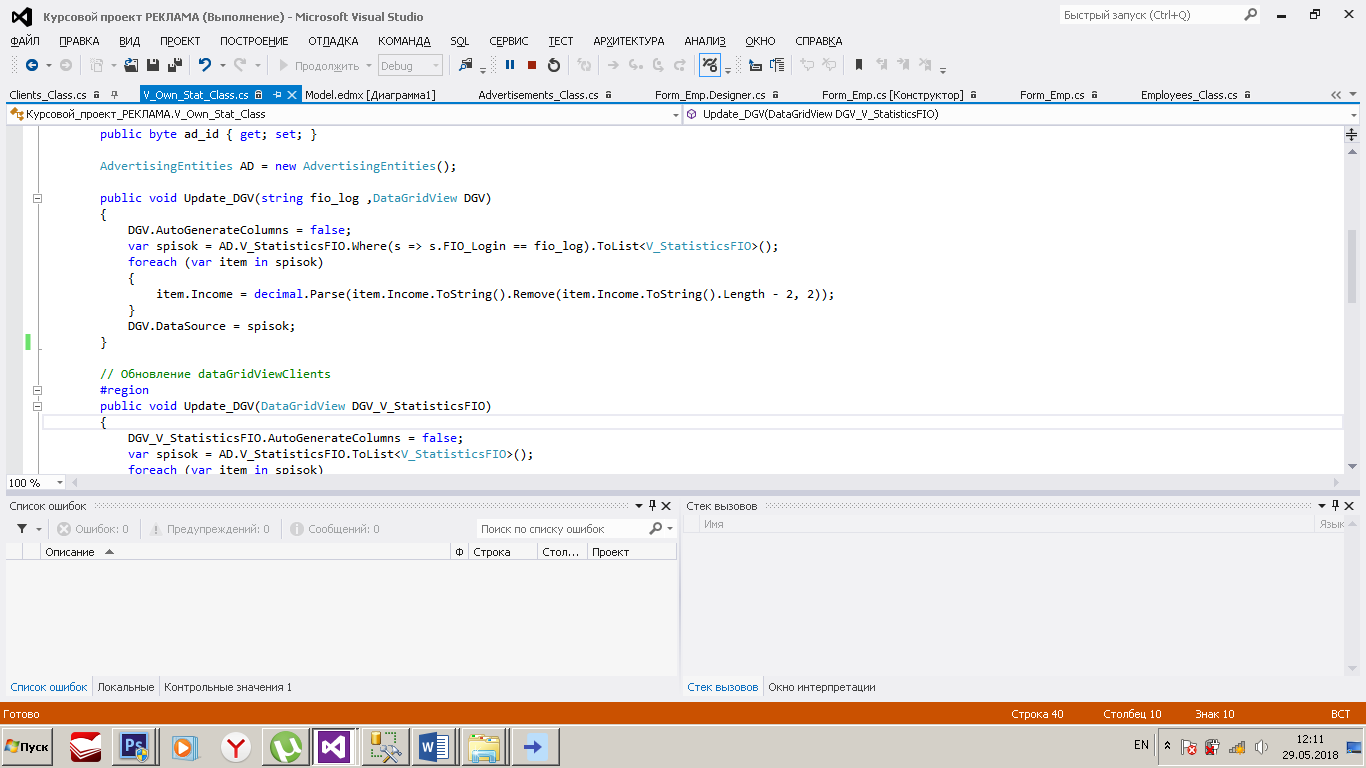


Рисунок 37 – метод для вывода всех сделок сотрудника в dataGridView, реализованный в классе.

На рисунке 37 происходит обращение к представлению «V\_StatisticFIO» для заполнения таблицы. На рисунке 36 суммируются значения прибыли каждой сделки.

## 2.6. Ведение информации о сотрудниках

Рассмотрим вариант, когда глава отдела входит в систему.

Ему предоставляется форма (рисунок 38), где расположены все необходимые для его деятельности функции.

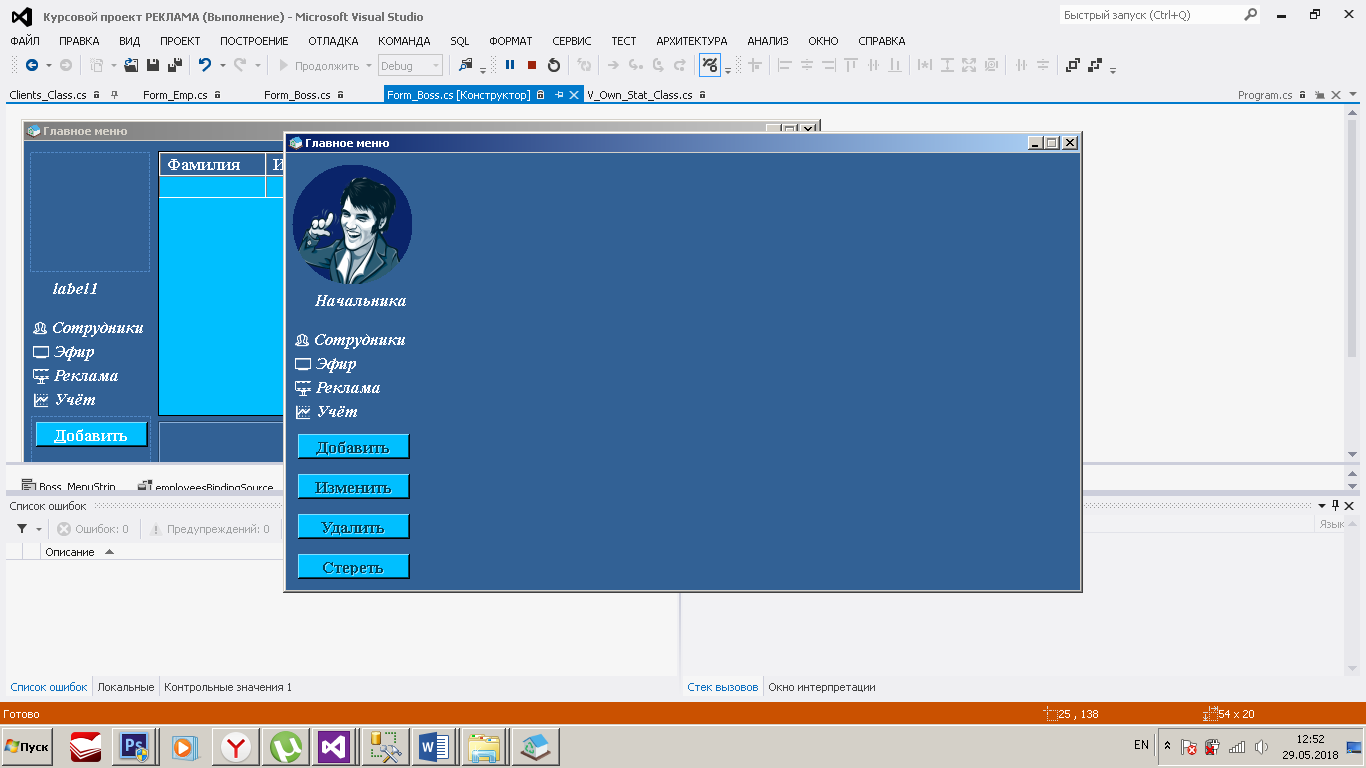


Рисунок 38 – форма «Главное меню» для главы отдела.

Форма имеет следующие объекты:

* изображение пользователя, которое размещено в элементе управления PictureBox;
* основное меню;
* кнопки для манипулирования с данными.

При выборе меню «Сотрудники» загружается dataGridView, который содержит в себе информацию о сотрудниках, хранимых в базе (рисунок 39).

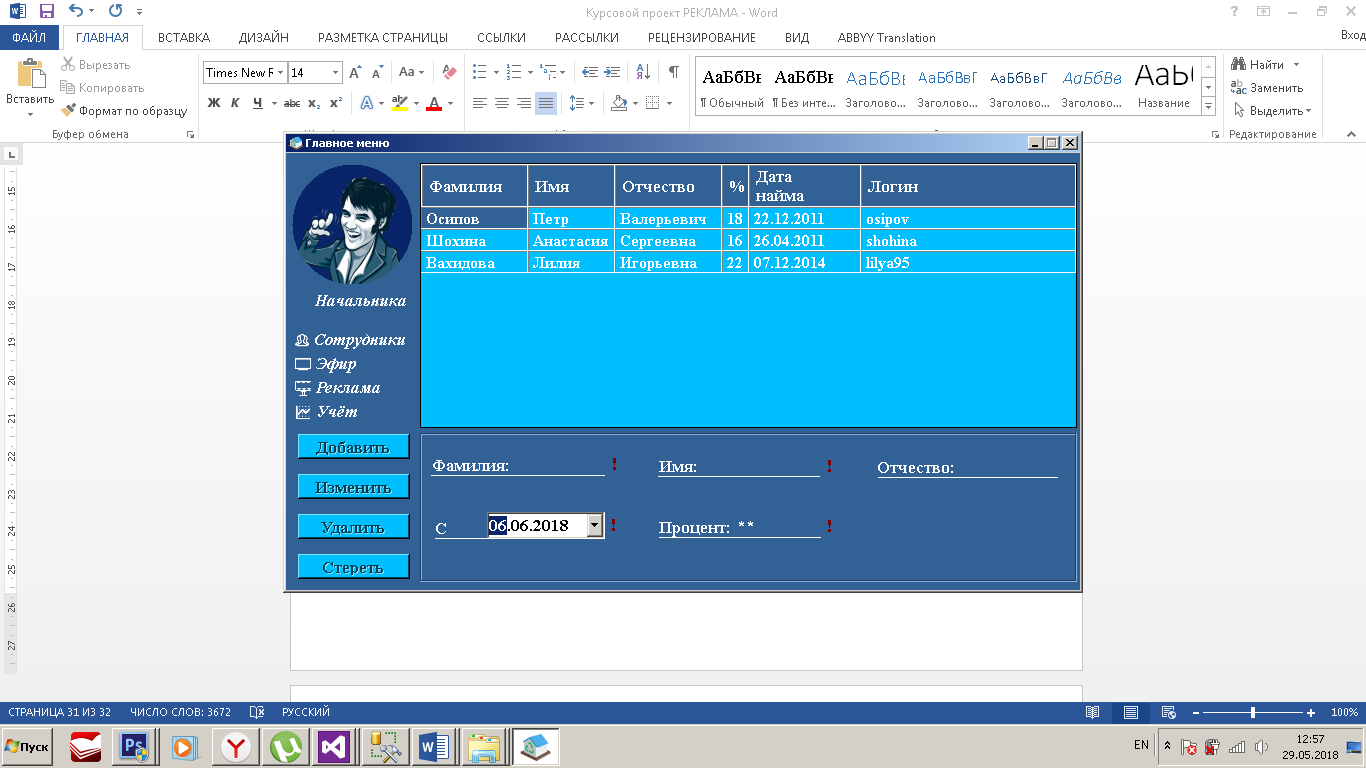


Рисунок 39 – Список сотрудников.

При двойном нажатии по записи в таблице, панель автоматически заполняется соответствующими данными (рисунок 40). После чего пользователю доступно две функции: удалить сотрудника или изменить его данные. Код для функции удаления представлен на рисунках 41-42, а для функции изменения – на рисунках 43-44.

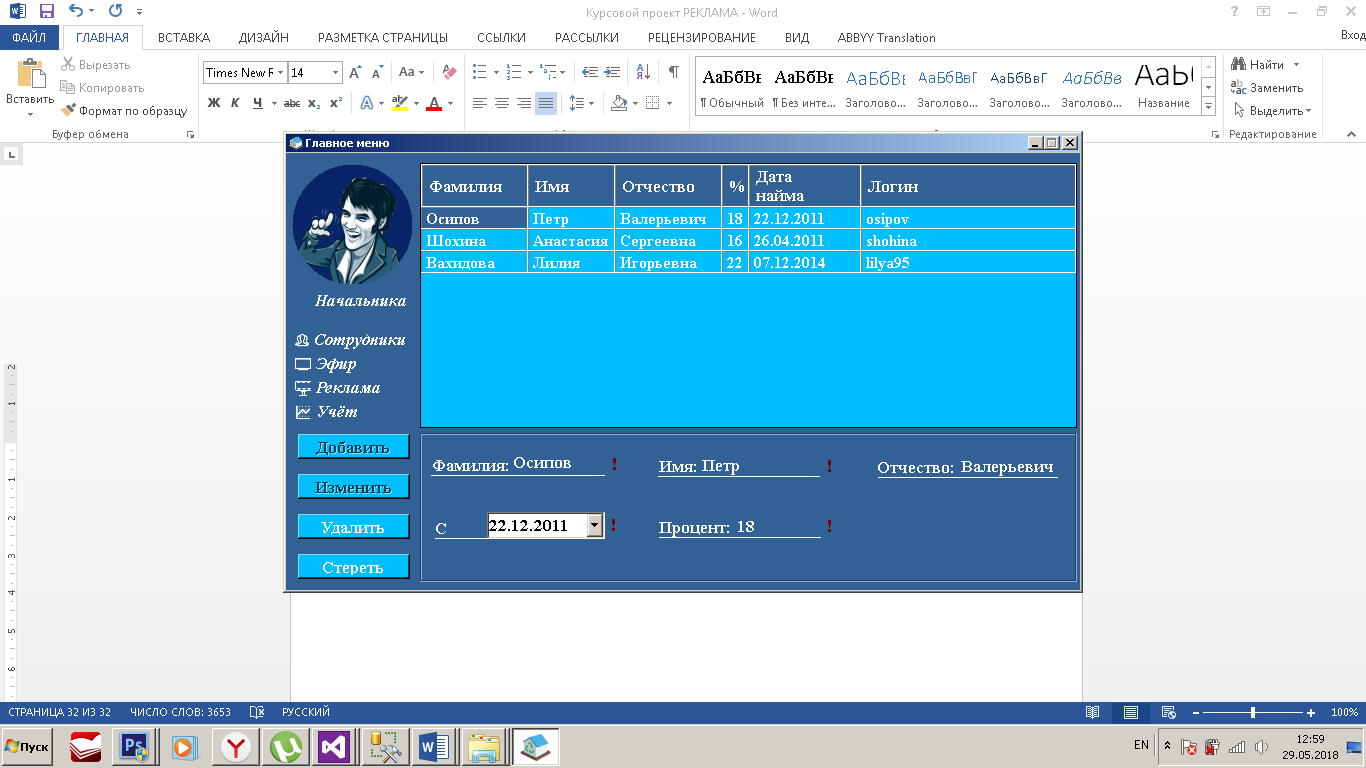


Рисунок 40 – Автозаполнение данными сотрудниками.

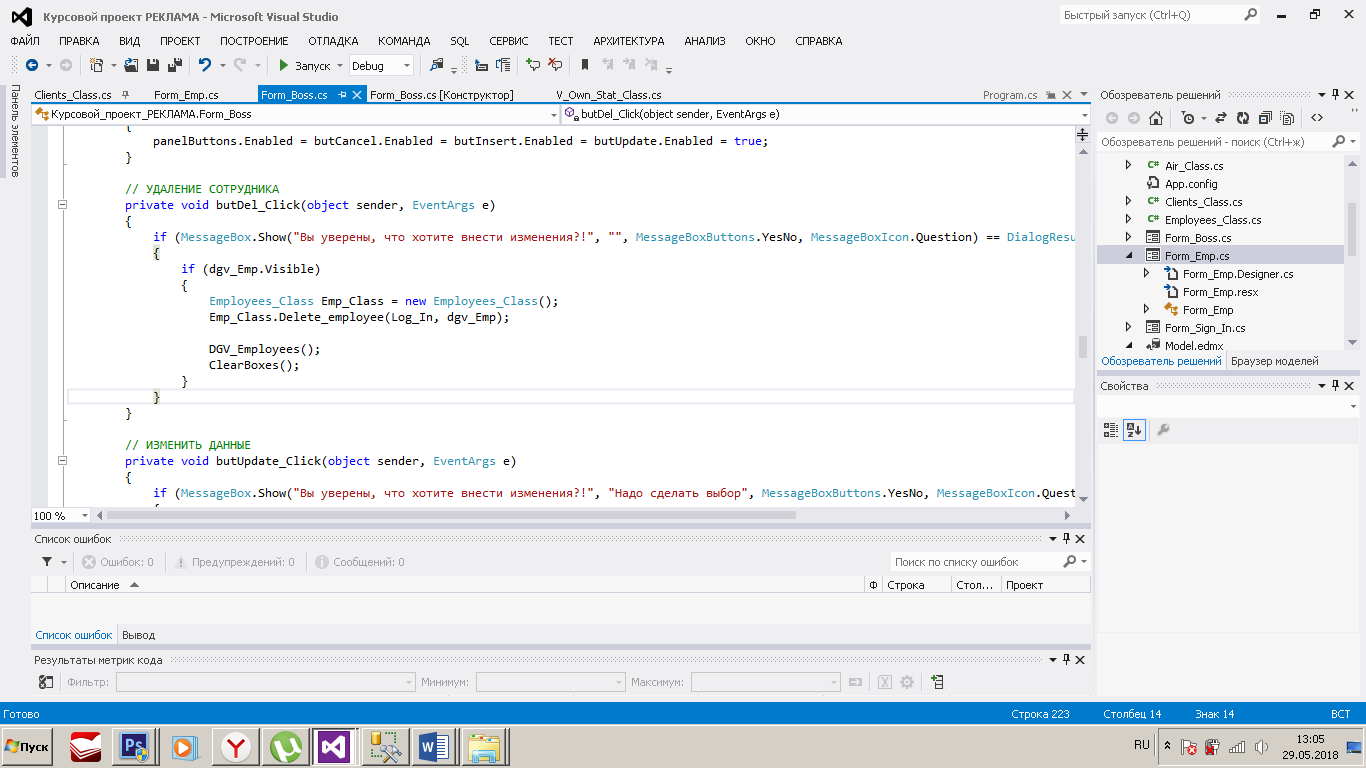


Рисунок 41– вызов метода для удаления сотрудника в обработчике события кнопки «Удалить».

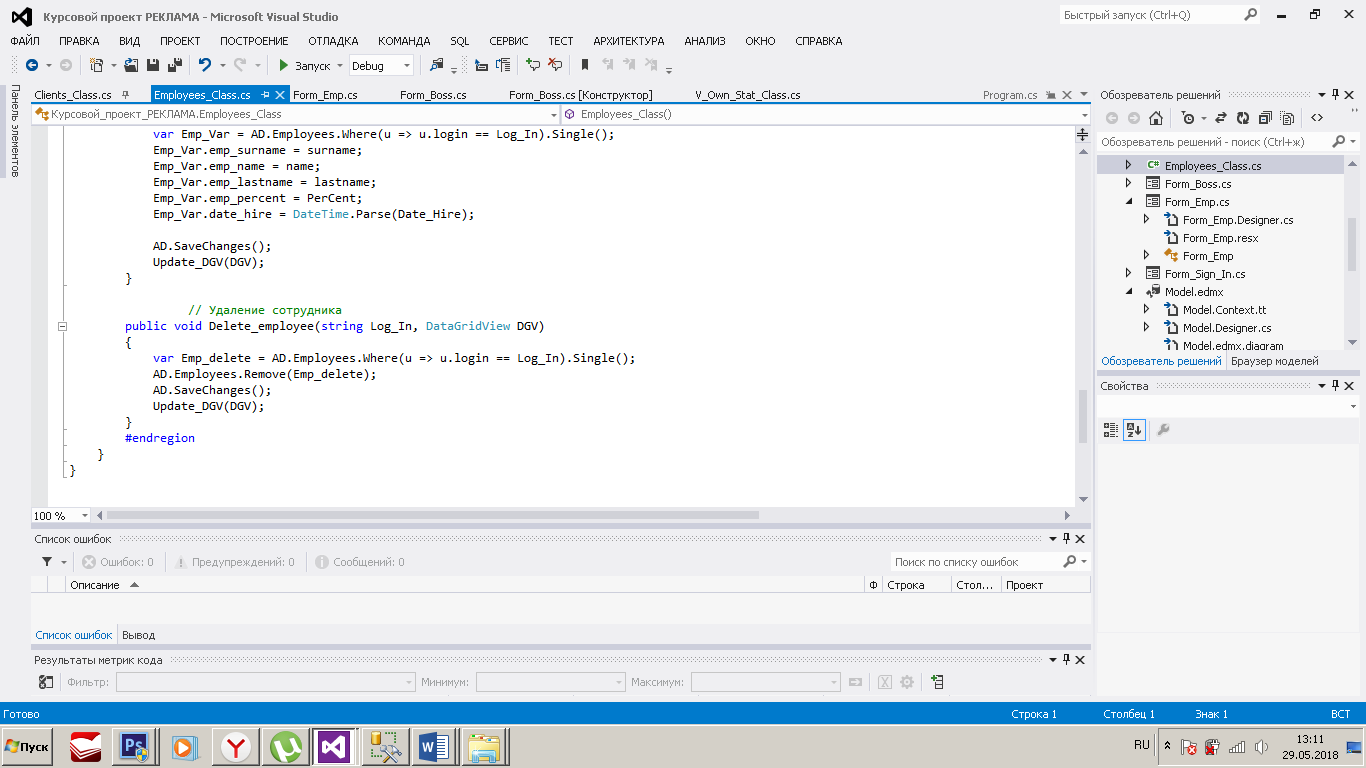


Рисунок 42– метод удаления сотрудника, реализованный в классе.

Алгоритм удаления сотрудника аналогичен алгоритму удаления клиента из базы.

После внесений изменений в заполненные записи на форме, пользователь может либо изменить действующего сотрудника (рисунок 43), нажав на кнопку «Изменить», либо добавить нового (рисунок 46), нажав на кнопку «Добавить». Также пользователь может самостоятельно заполнить данные на форме, чтобы добавить сотрудника. Код реализации функции изменения данных о сотруднике представлен ниже на рисунках 44 – 45, а для добавления нового сотрудника – на рисунках 47 – 48.

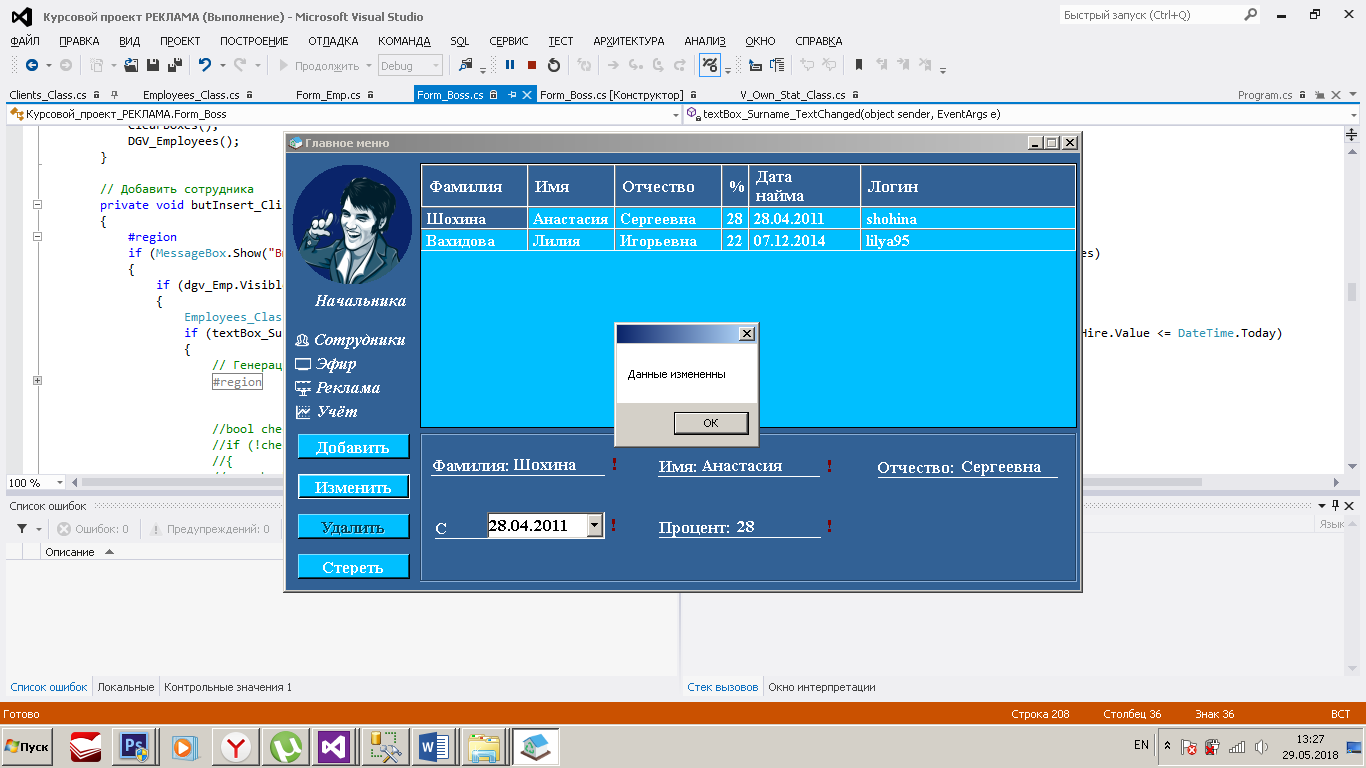


Рисунок 43 – результат после изменений данных о сотруднике.

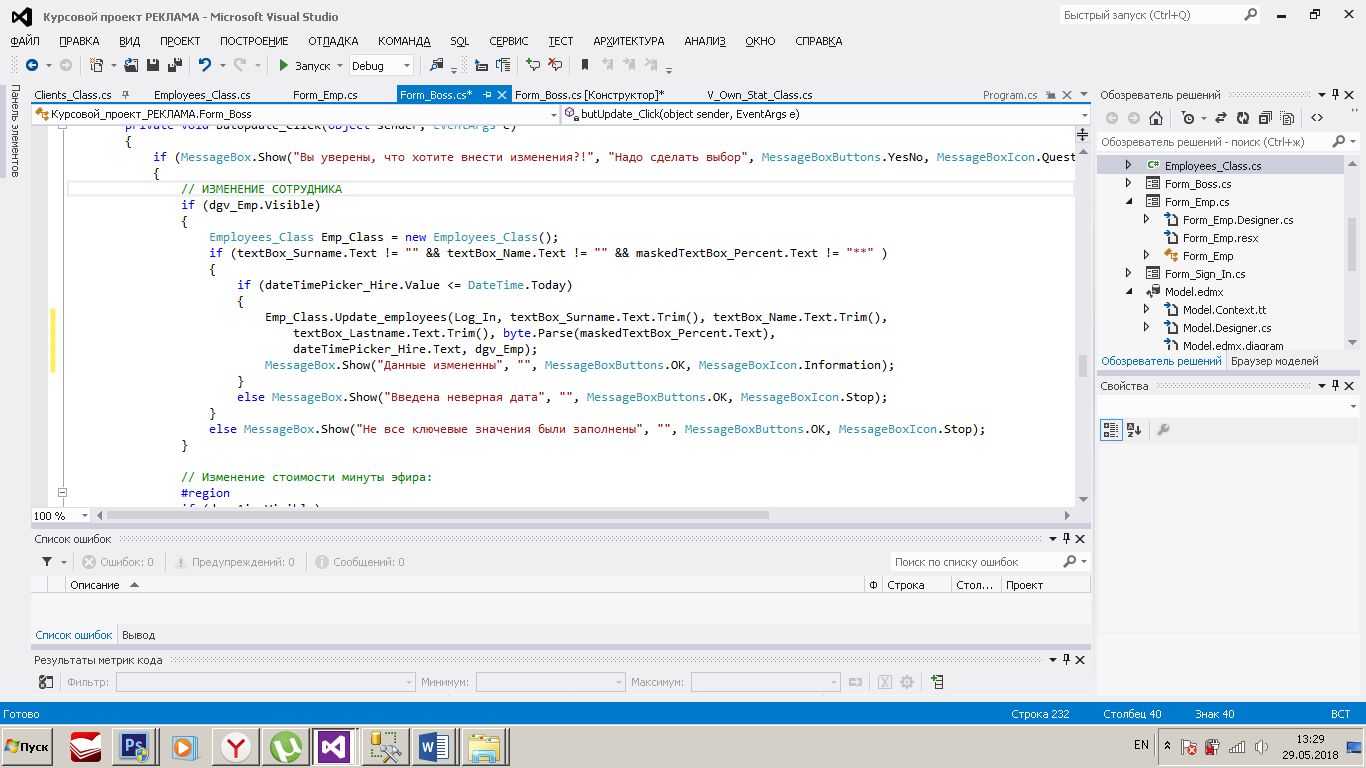


Рисунок 44– вызов метода для изменения данных сотрудника в обработчике события кнопки «Изменить».

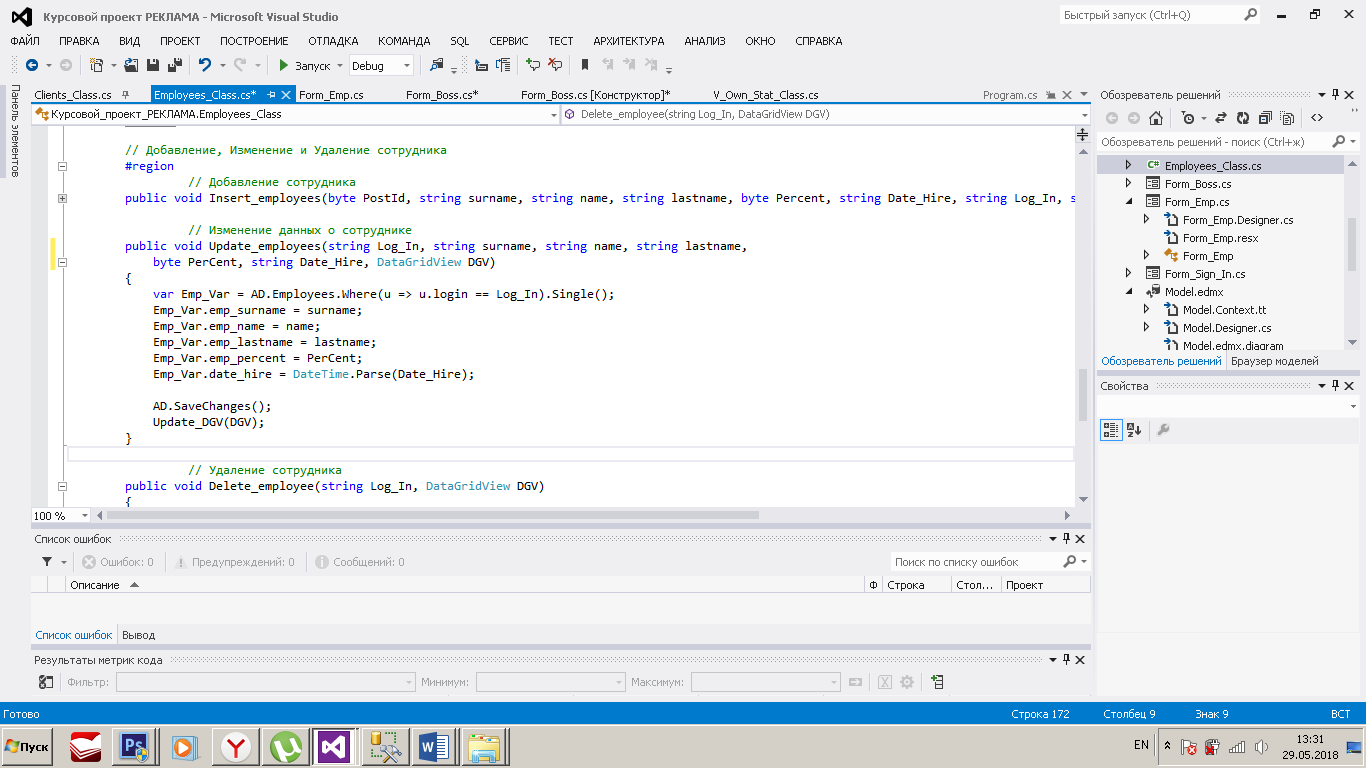


Рисунок 45– метод для изменения сотрудника, реализованный в классе.

Алгоритм изменения данных сотрудника аналогичен алгоритму изменения данных клиента.

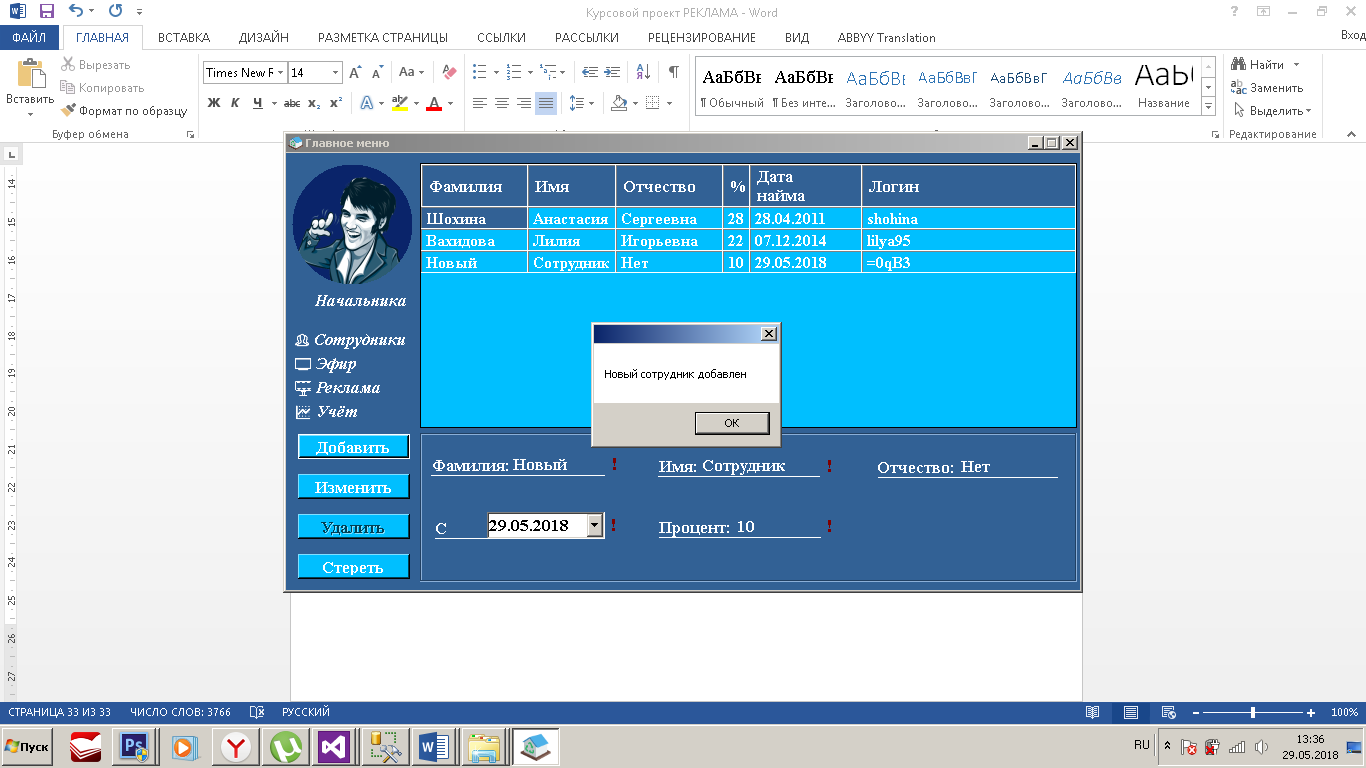


Рисунок 46 – результат добавления нового сотрудника.

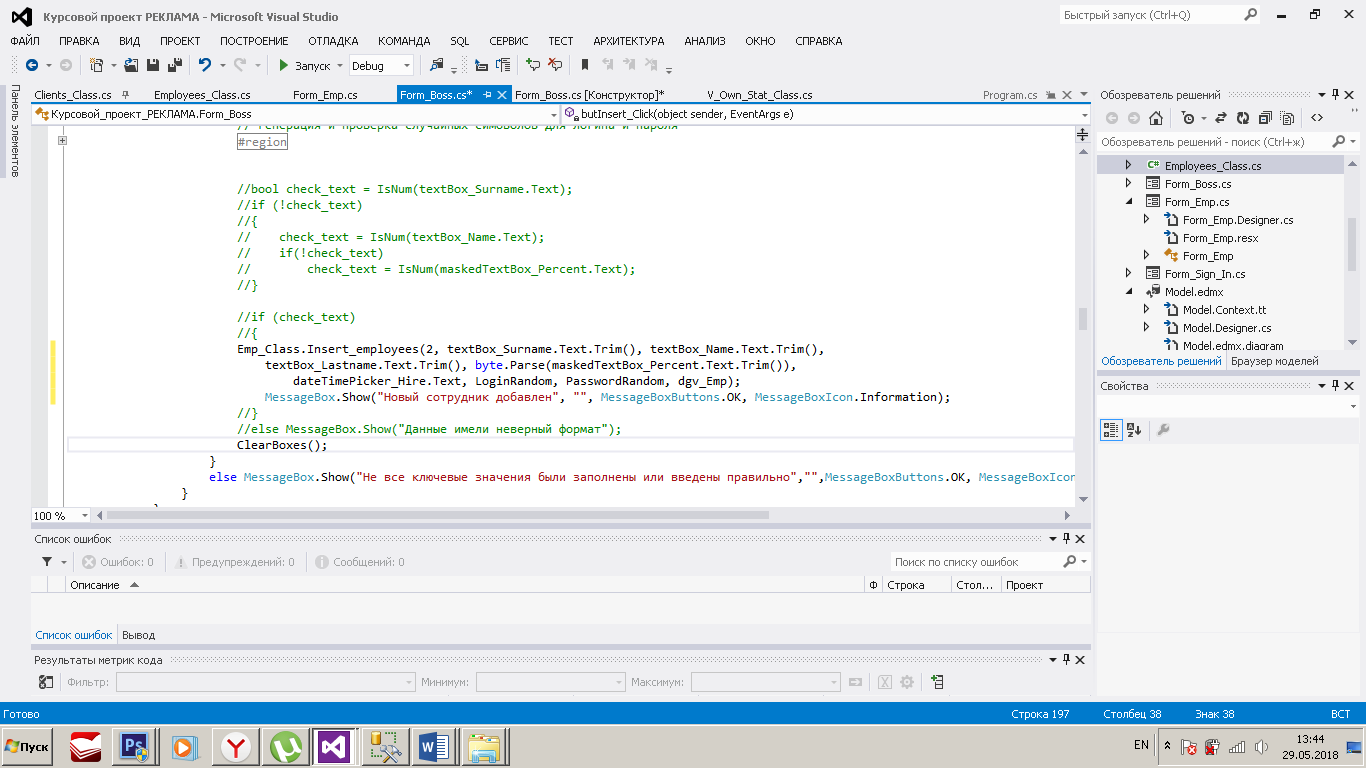


Рисунок 47– вызов метода для добавления сотрудника в обработчике события кнопки «Добавить».

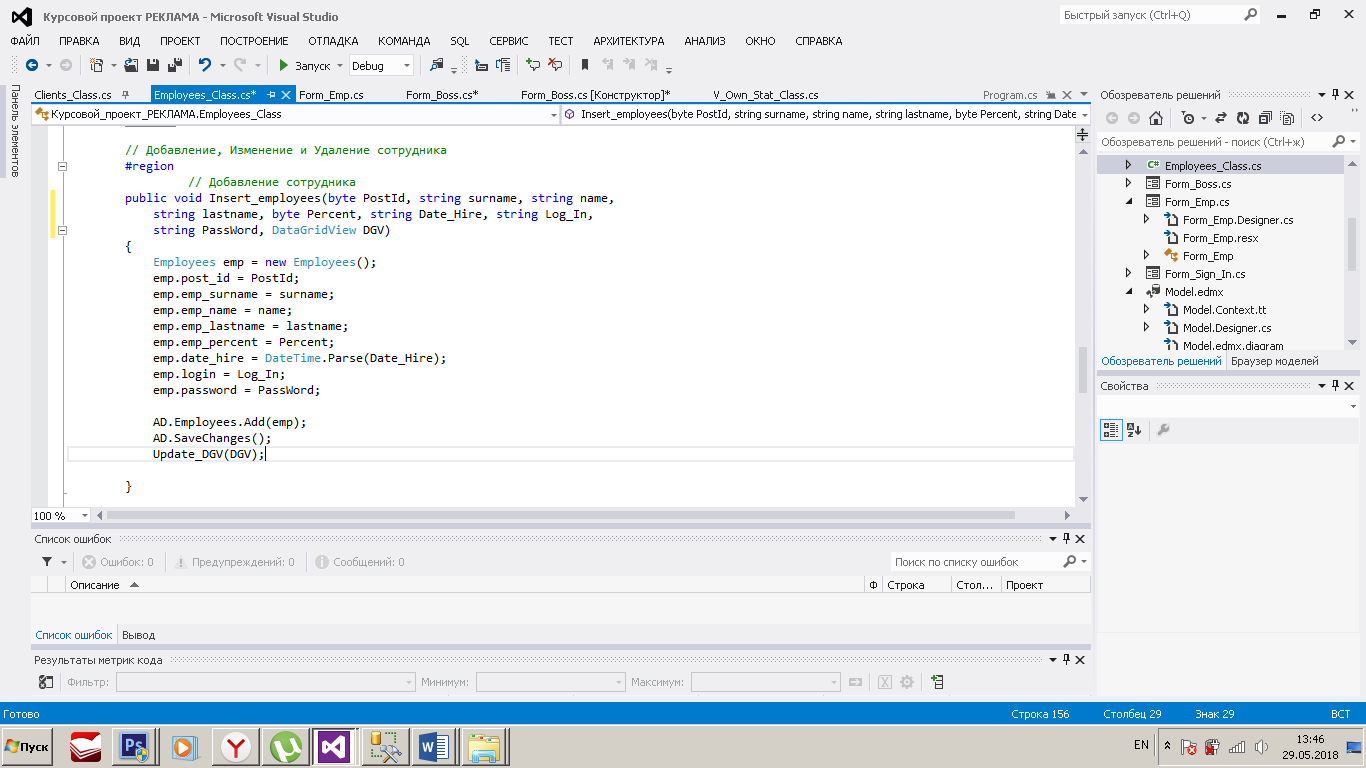


Рисунок 48– метод для добавления сотрудника, реализованный в классе.

Алгоритм добавления нового сотрудника аналогичен алгоритму добавления клиента.

## 2.7. Изменение стоимости за рекламу

Для изменения стоимости за услугу, глава отдела должен зайти в меню «Эфир». Далее он выбирает определённую передачу из списка и задаёт новое значение стоимости (рисунок 49). После нажимает на кнопку «Изменить» (рисунок 50).

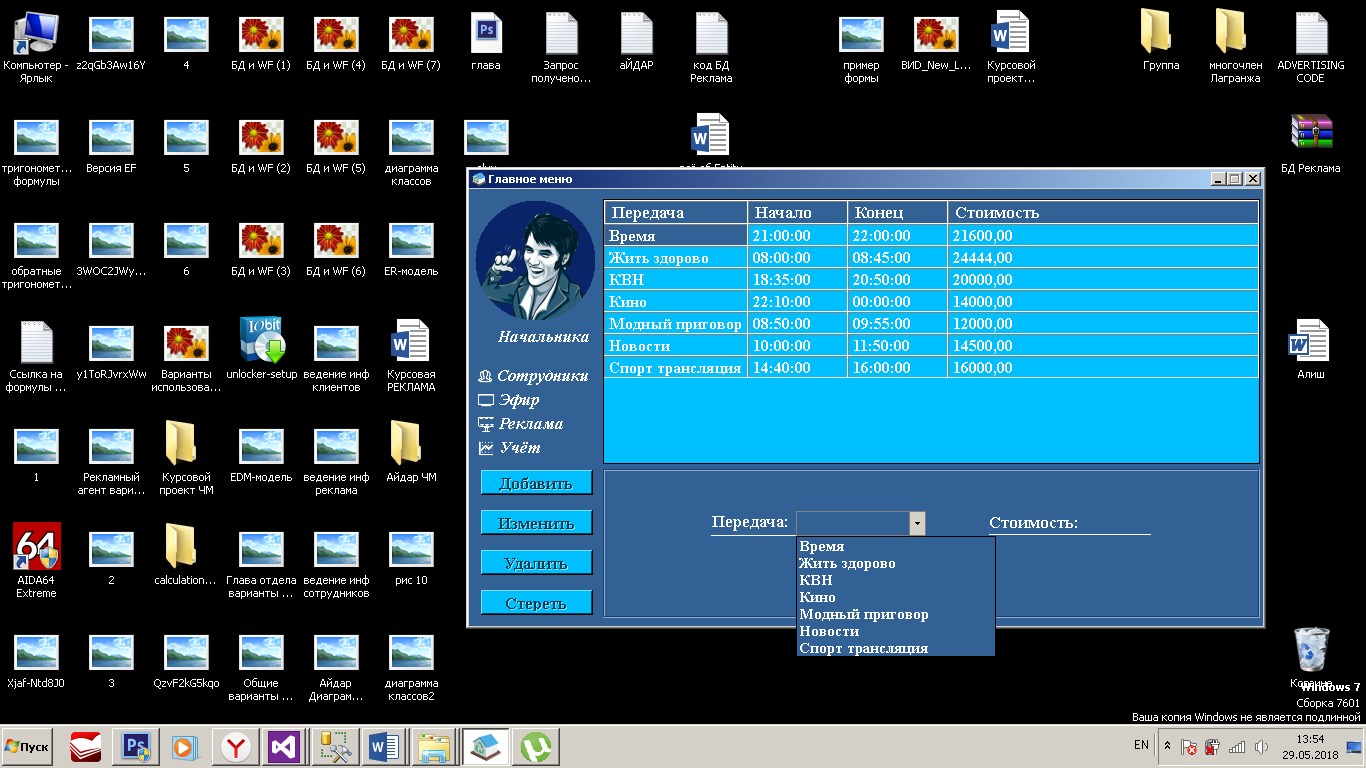


Рисунок 49 – меню «Эфир».

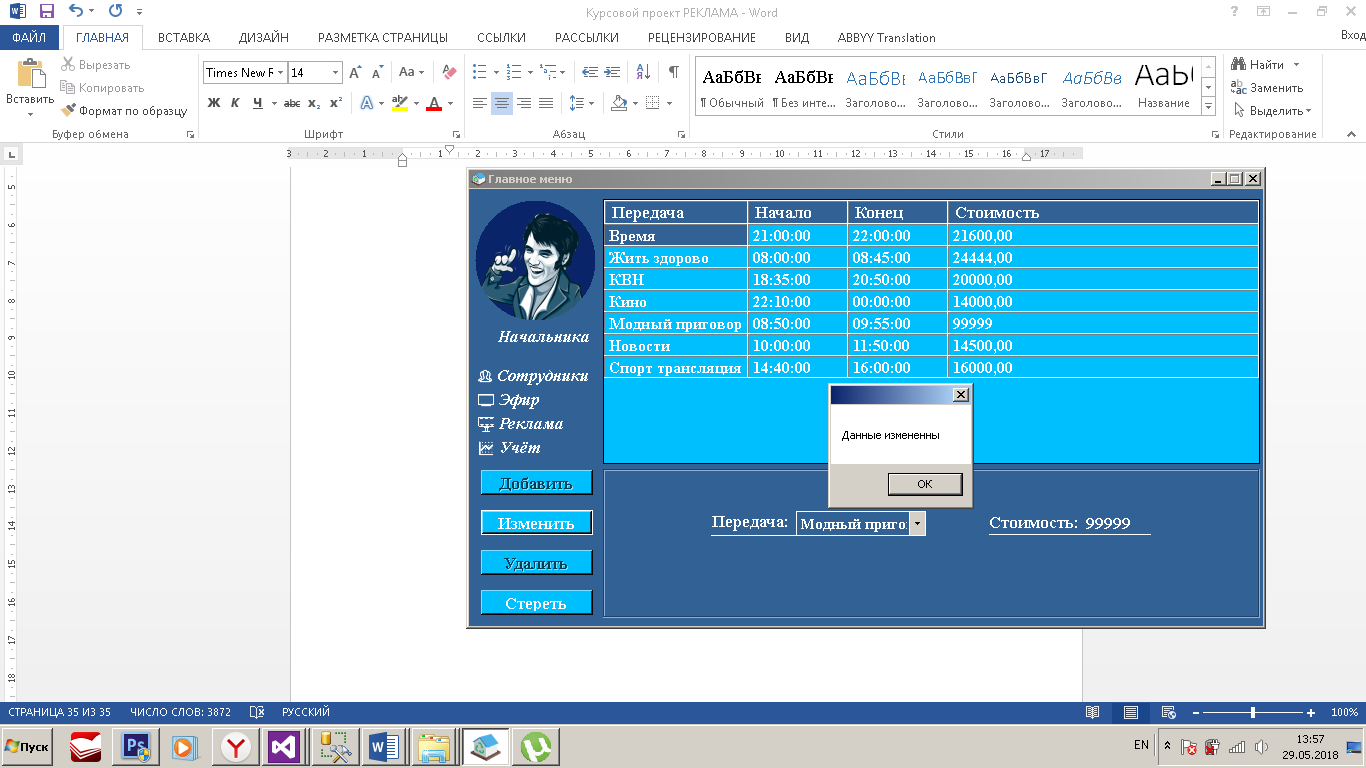


Рисунок 50 – результат изменения стоимости за рекламу.

Код для реализации данной функцией представлен на рисунках 51 – 52.

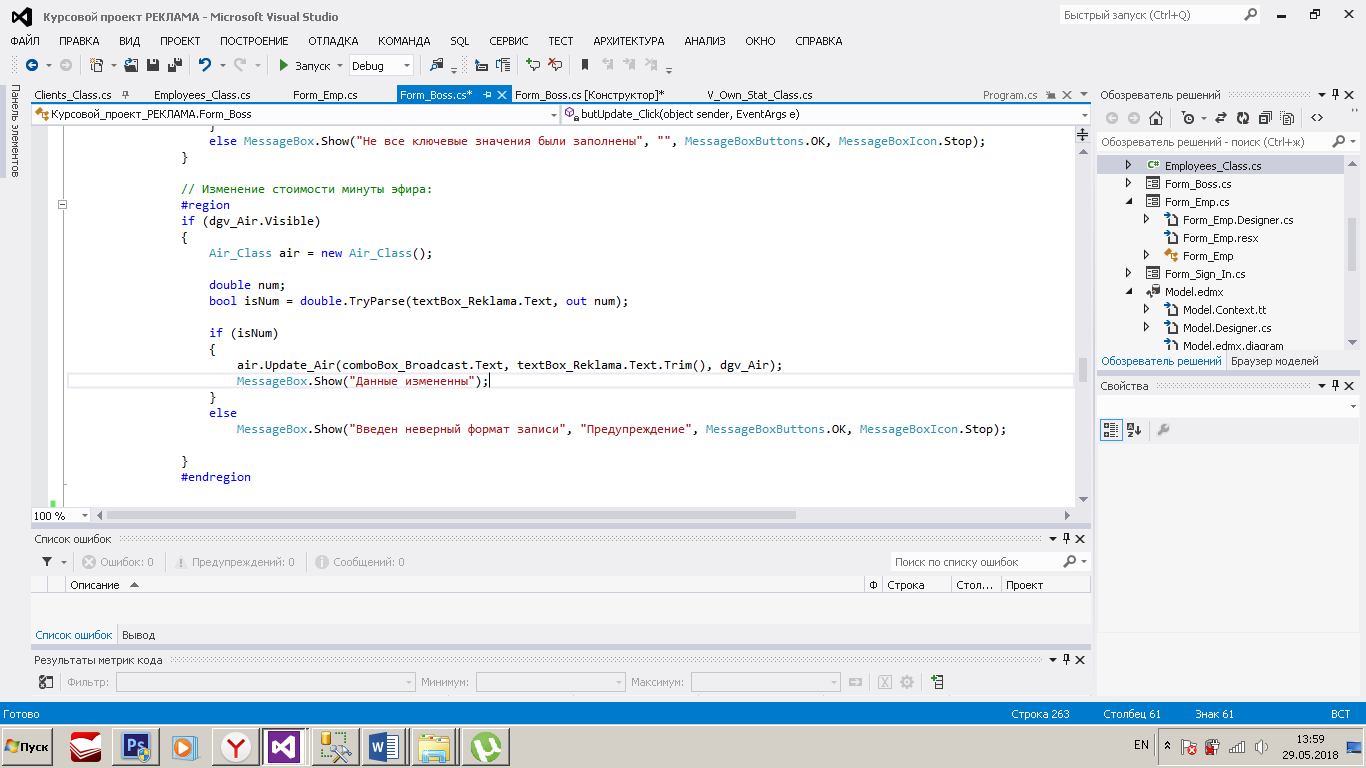


Рисунок 51– вызов метода для изменения стоимости в обработчике события кнопки «Изменить».

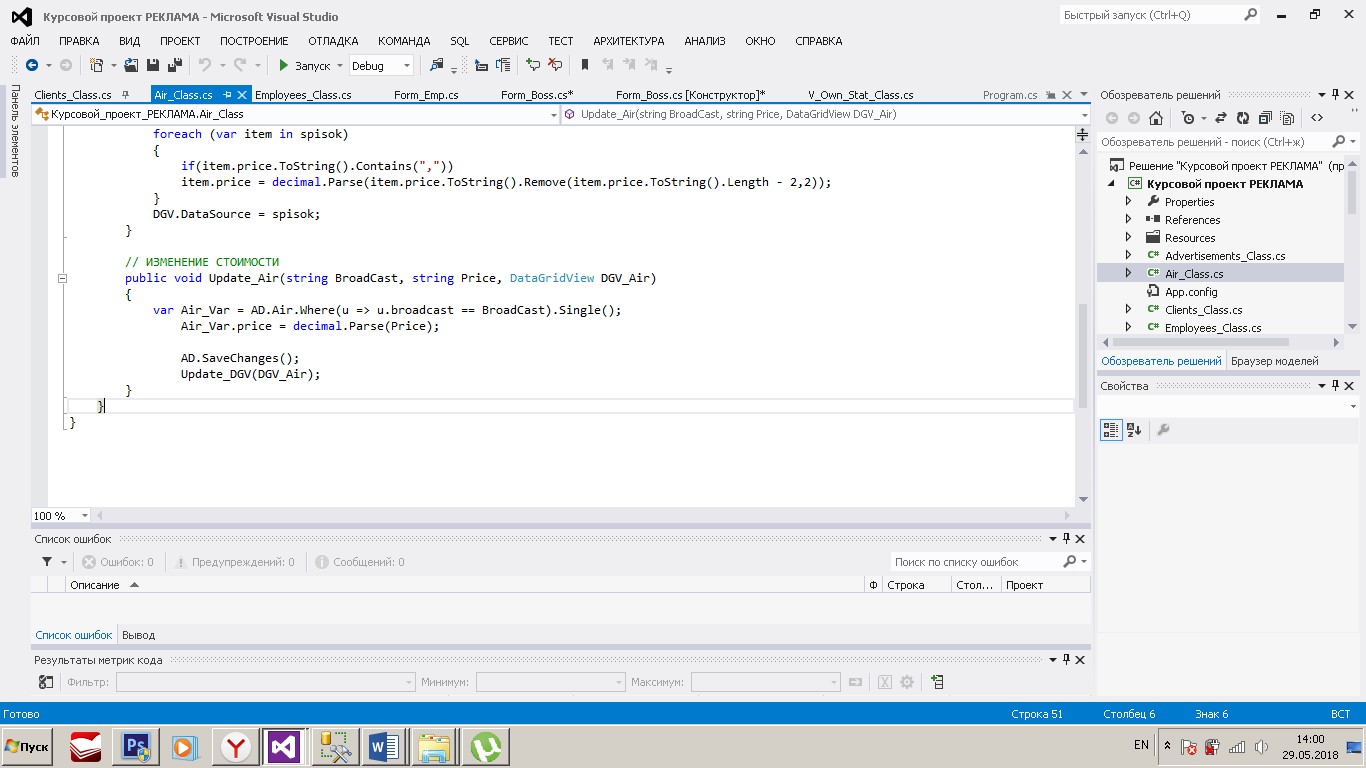


Рисунок 52– метод для изменения стоимости, реализованный в классе.

На рисунке 51 также имеется метод для проверки назначения стоимости, так как стоимость может иметь только числовое выражение.

Алгоритм изменения стоимости за рекламу аналогичен алгоритму изменения сотрудника.

## 2.8. Просмотр статистики

Для просмотра статистики глава отдела должен зайти в меню «Учёт».

Появляется таблица со всеми сделками с данными клиентов. Далее пользователь выбирает определённого сотрудника из списка. После нажимает на кнопку «Поиск» (рисунок 53).

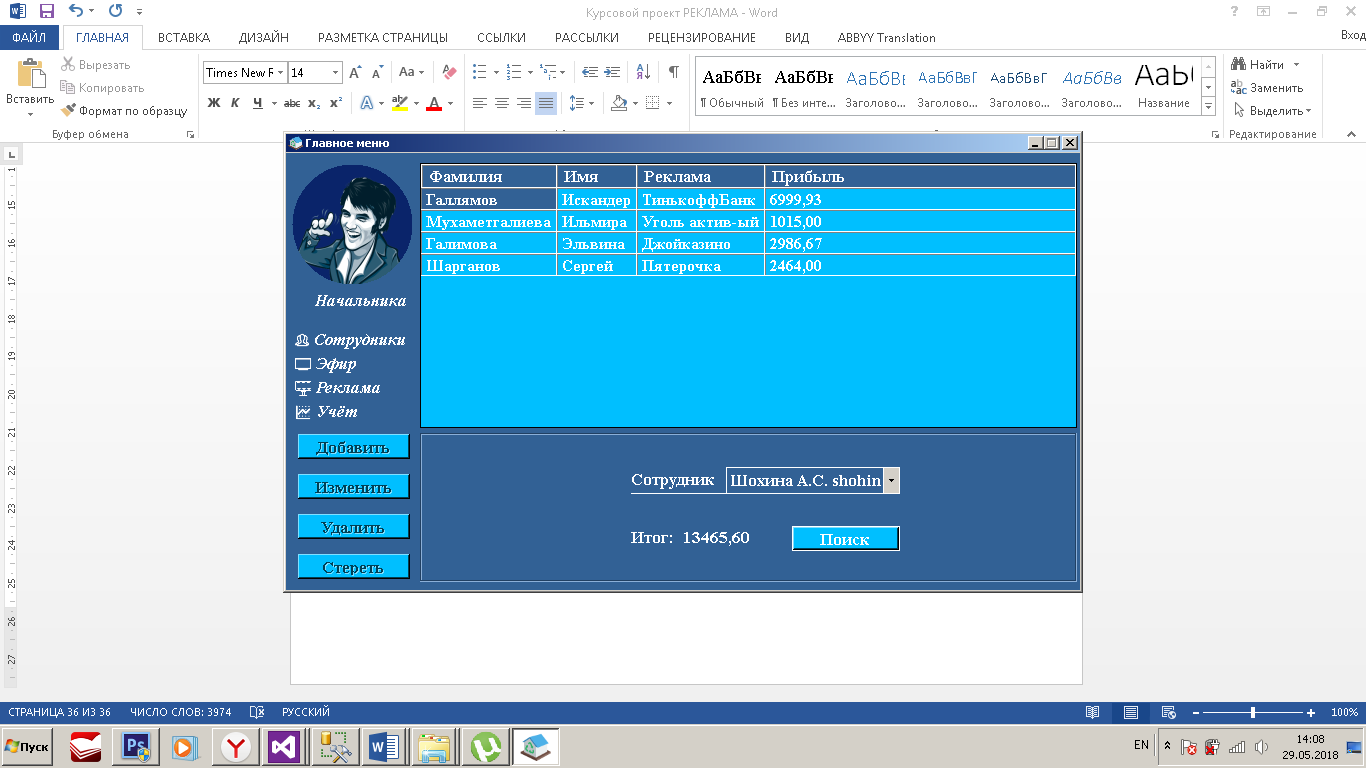


Рисунок 53 – просмотр статистики отдельного сотрудника.

На рисунке 54 представлен код для заполнения списка сотрудников.

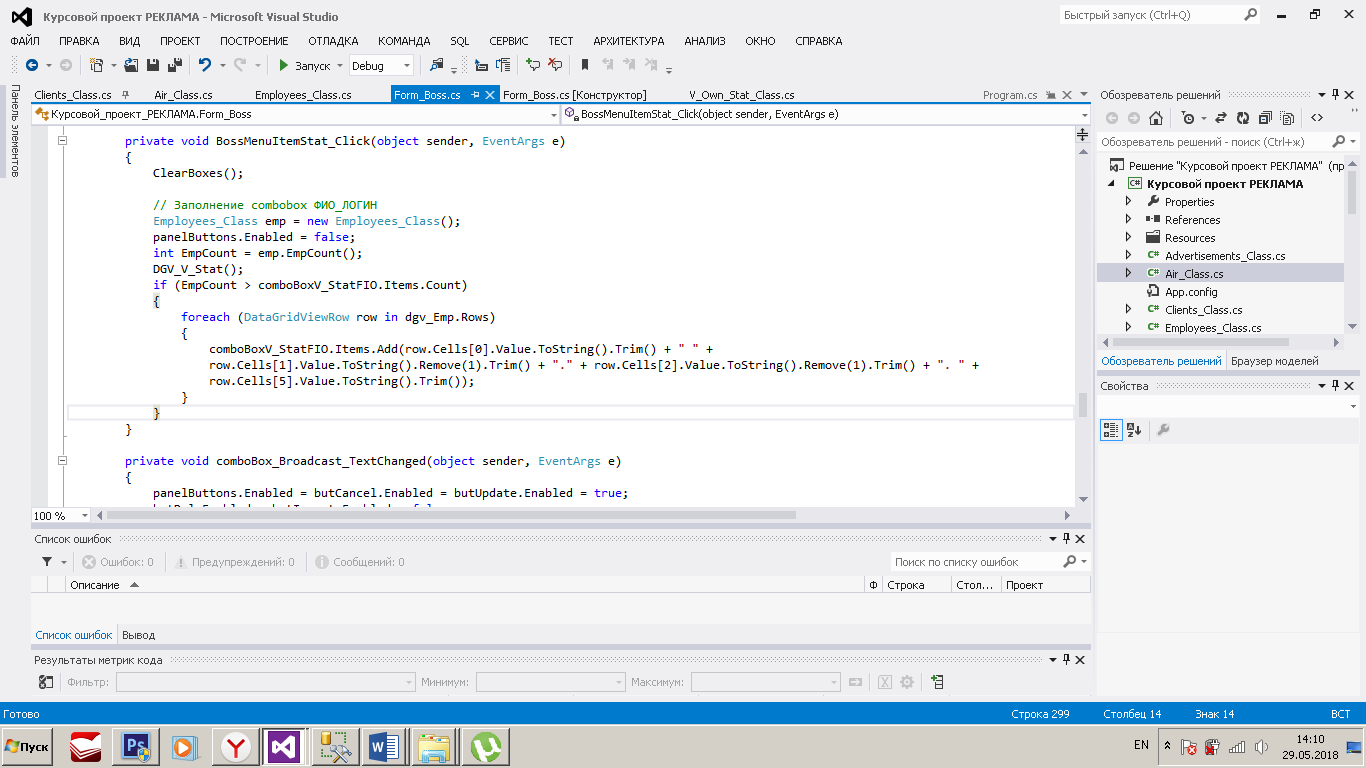


Рисунок 54 – код для заполнения выпадающего списка сотрудников.

На рисунке 55 представлен код для кнопки «Поиск».

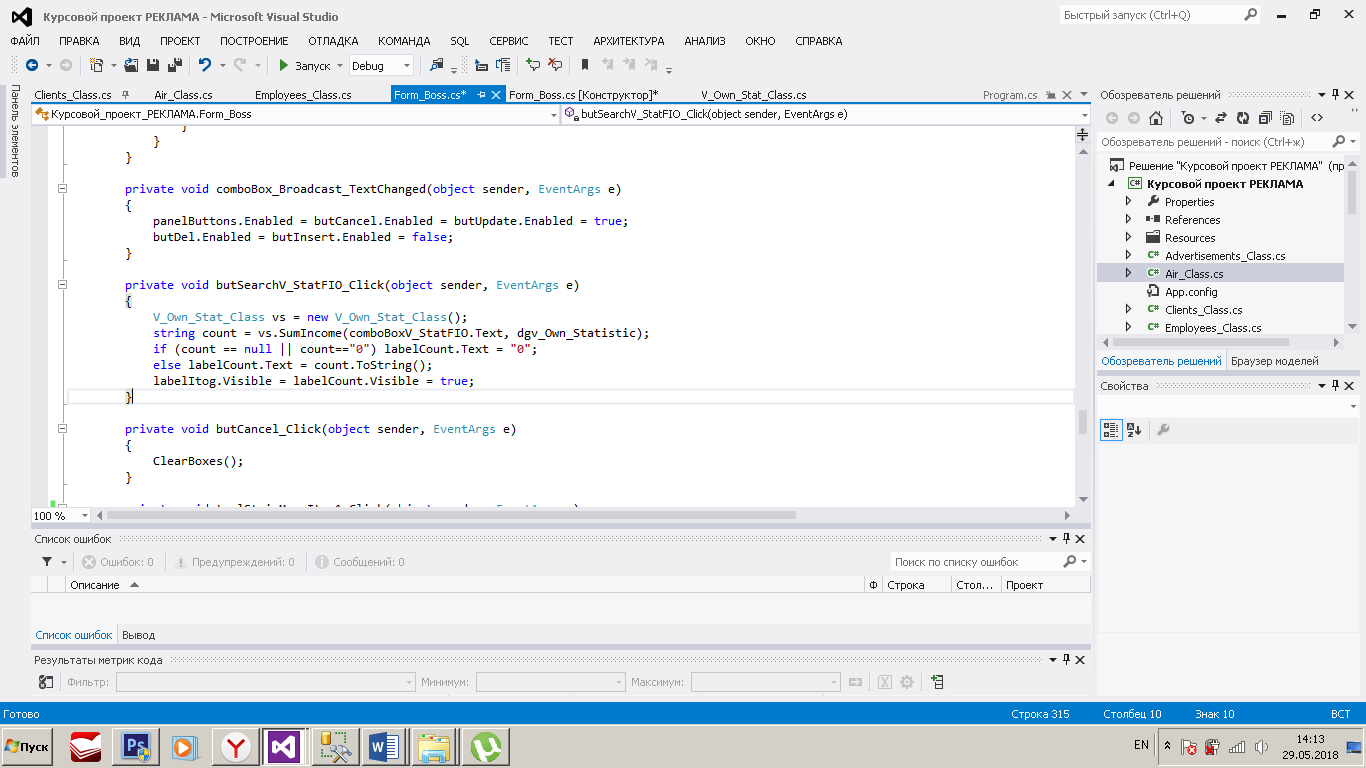


Рисунок 55 – вызов метода для подсчёта дохода сотрудника в обработчике событий на кнопке «Поиск».

На рисунке 56 представлен метод, реализованный в классе, который подсчитывает прибыль выбранного сотрудника из списка за все его сделки.

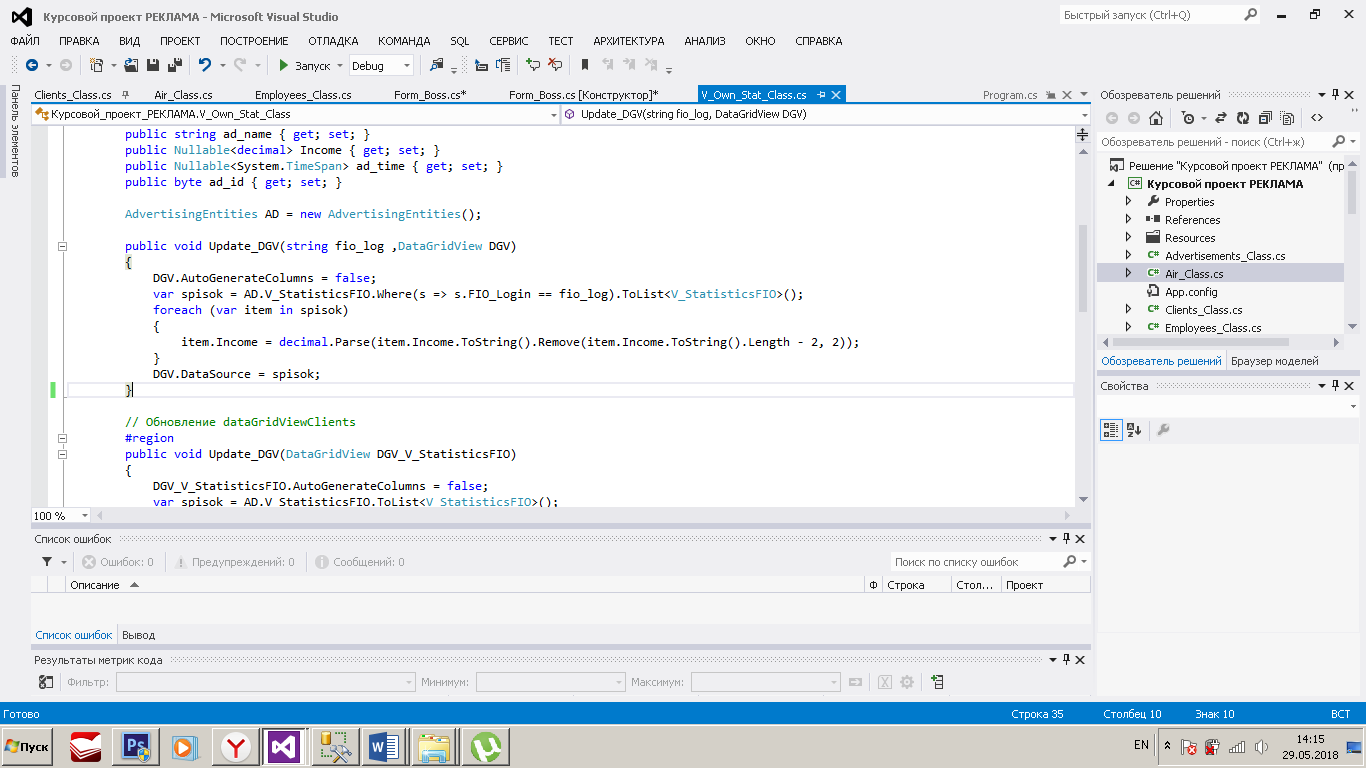


Рисунок 56 – метод для подсчёта прибыли сотрудника, реализованный в классе.

## 2.9. Выход из системы

Для того, чтобы пользователь вышел из системы, ему необходимо вернуться в окно с авторизацией, щёлкнув правой кнопкой мыши по изображению и выбрать элемент «Выход» (рисунок 57), после чего, согласившись на сообщение, система открывает окно с авторизацией, остаётся только нажать на кнопку «Выйти» и ответить на сообщение (рисунок 58).

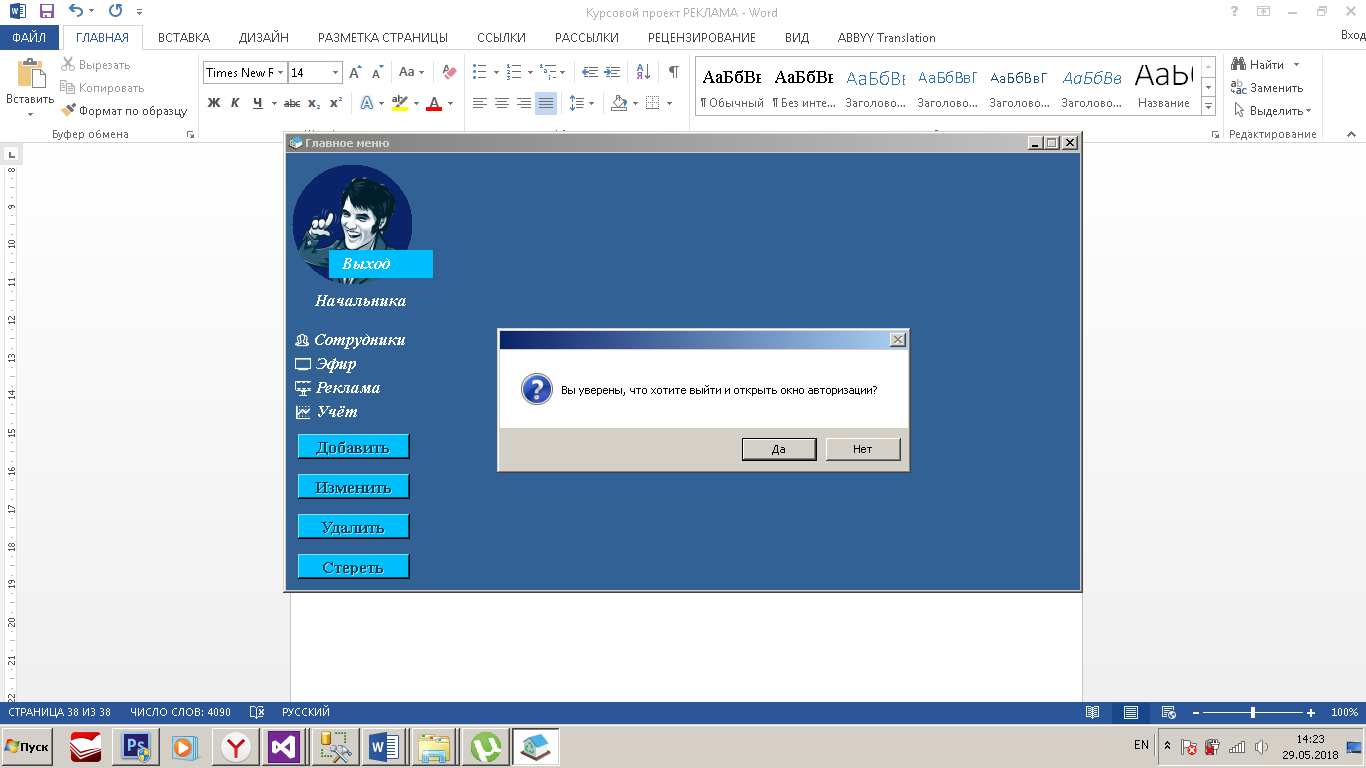


Рисунок 57 – элемент «Выход».

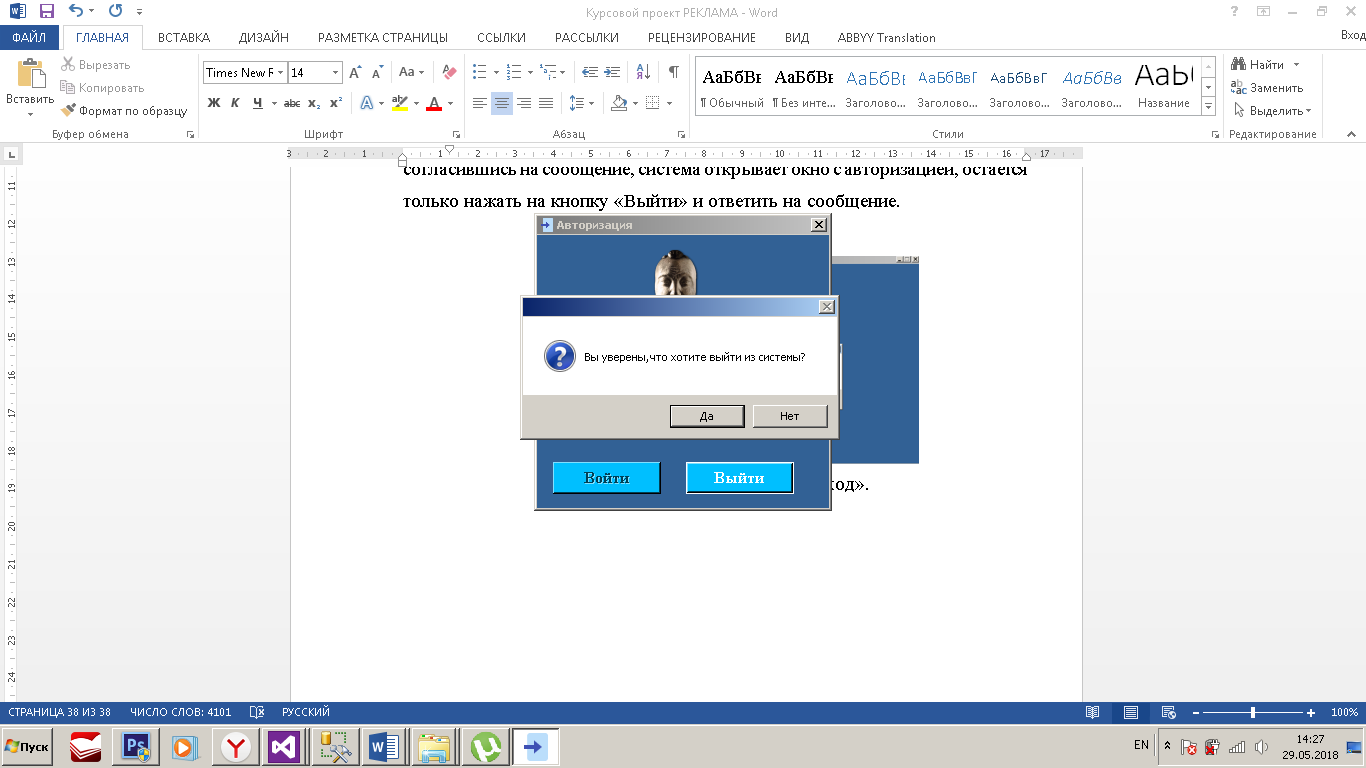


Рисунок 58 – выход из системы.

На рисунке 59 представлен код, который срабатывает при любом закрытии формы «Главное меню».

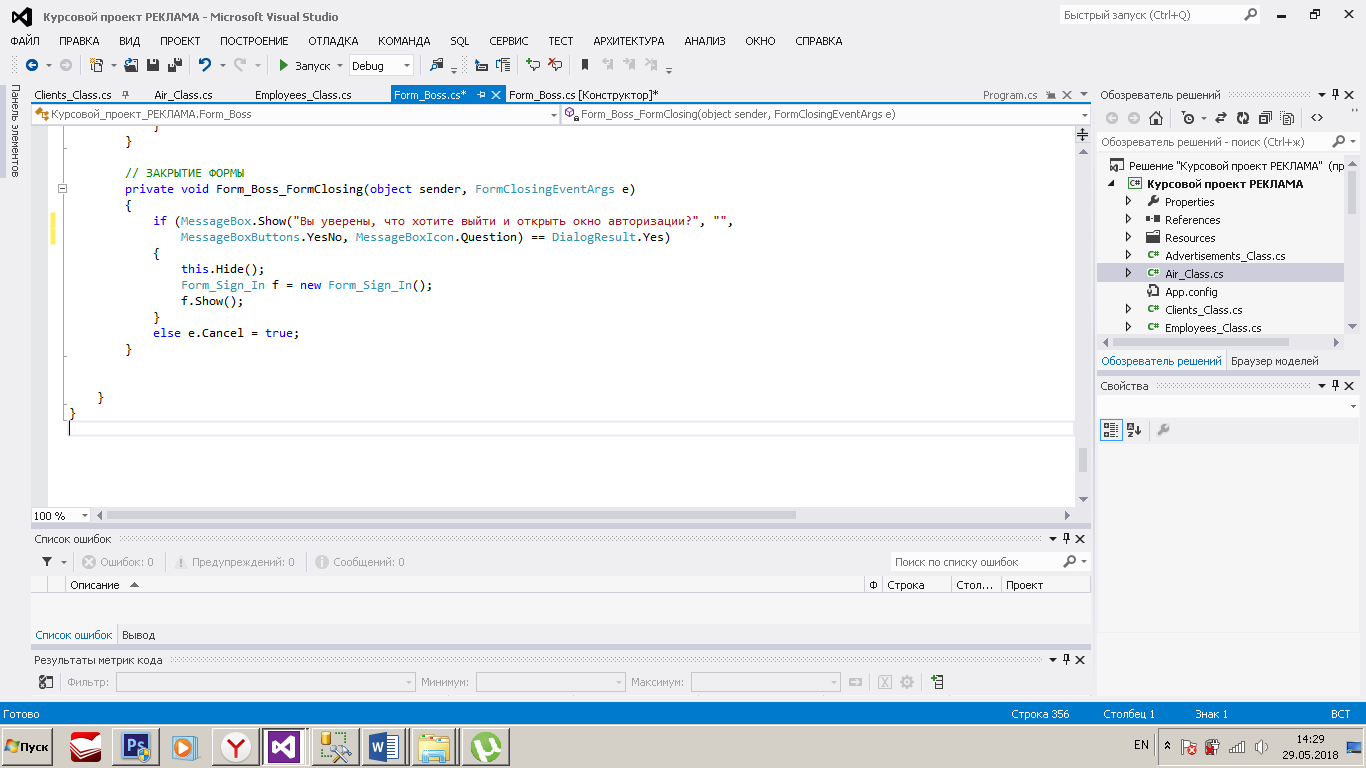


Рисунок 59 – событие для реализации выхода в окно с авторизацией.

На рисунке 60 представлен код, который срабатывает при любом закрытии формы «Авторизация».

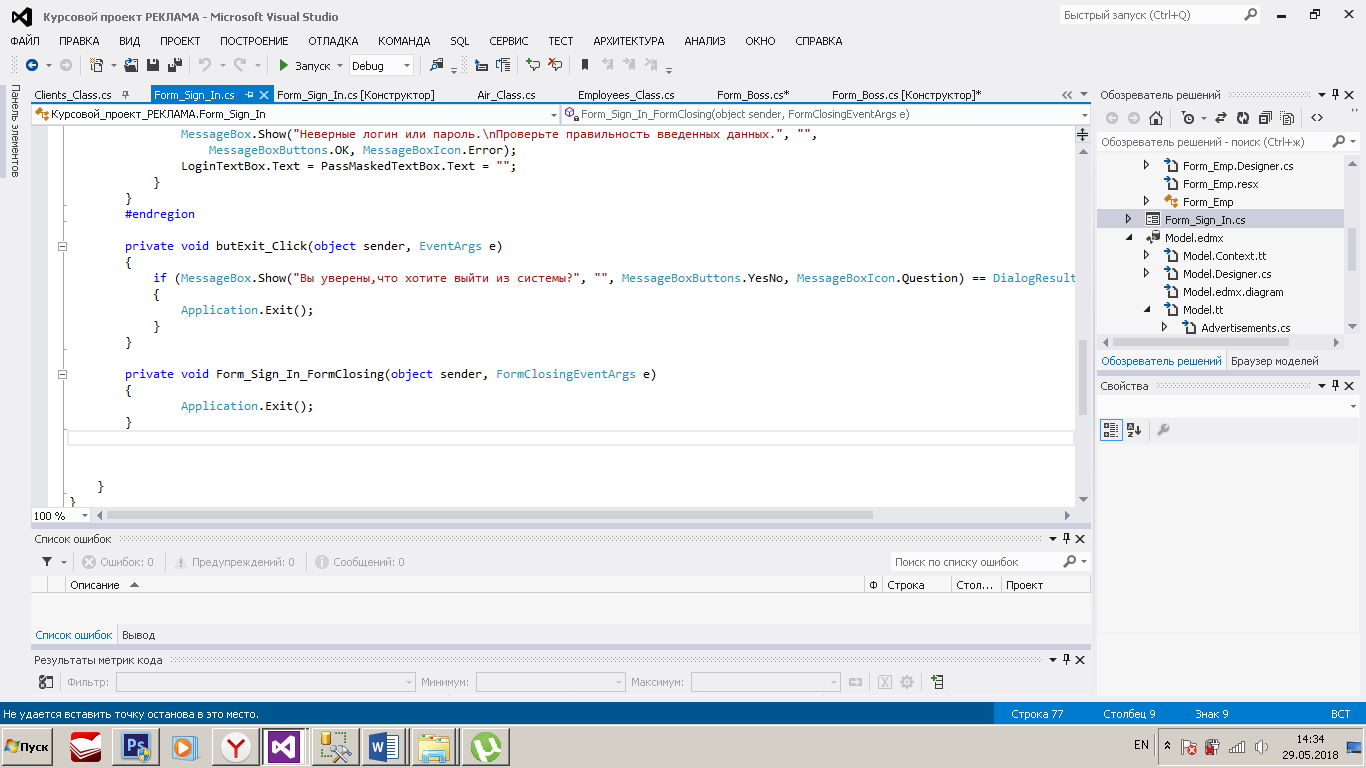


Рисунок 60 – закрытие приложения.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения данного курсового проекта был разработан прототип информационной системы «Учет телекомпанией стоимости прошедшей в эфире рекламы», которая обеспечивает информационную поддержку деятельности рекламной службы телеканала.

Преимущество использования данной системы состоит в том, что приложение обладает оперативностью обработки, приятным и удобным интерфейсом.

Также была изучена технология Entity Framework с использованием простых и лаконичных лямбда – выражений, которые позволяют заменить стандартные запросы, сформированные на языке SQL.

## СПИСОК ЛИТЕРАТУРЫ

1. Бодров О. А., Медведев Р. Е. Предметно-ориентированные экономические информационные системы. - М.:Горячая линия- Телеком, 2013. - 244 с.
2. Редмонд, Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL, М.: ДМК Пресс, 2013. – 384 с.
3. Робинсон, Я. Графовые базы данных: новые возможности для работы со связанными данными [Электронный ресурс] / Я. Робинсон, Д. Вебер, Э. Эифрем. – Электрон. дан. – Москва : ДМК Пресс, 2016. – 256 с.
4. Осипов, Д.Л. nterBase и Delphi. Клиент-серверные базы данных [Электронный ресурс] – Электрон. дан. – Москва : ДМК Пресс, 2015. – 536 с.
5. Джонатан, Л. Ядро Oracle. Внутреннее устройство для администраторов и разработчиков баз данных [Электронный ресурс] – Электрон. дан. – Москва : ДМК Пресс, 2015. – 372 с.
6. Латыпова, Р.Р. Базы данных. Курс лекций : учебное пособие / Р.Р. Латыпова. - Москва : Проспект, 2016. – 96 с.
7. Агальцов В.П. Базы данных. В 2-х кн. Книга 2. Распределенные и удаленные базы данных : учебник / В.П. Агальцов. – М. : ИД «ФОРУМ» : ИНФРА-М, 2017. – 271 с.
8. Шустова Л.И. Базы данных : учебник / Л.И. Шустова, О.В. Тараканов. – М. : ИНФРА-М, 2017. – 304 с.
9. Голицына, О.Л. Основы проектирования баз данных: учеб. пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. – 2-е изд., перераб. и доп. – М. : Форум : ИНФРА-М, 2017. – 416 с.
10. Кара-Ушанов, В.Ю. SQL — язык реляционных баз данных: учебное пособие, Издательство: Уральский федеральный университет, 2016. – 156 с.